CTE-STEM 2021

5TH APSCE INTERNATIONAL CONFERENCE ON COMPUTATIONAL THINKING AND STEM EDUCATION

Proceedings of International Teacher Forum

2nd - 4th June 2021

Organised by:

APSCE

Hosted by:

NIE National Institute of Education Singapore

An Institute of Nanyang Technological University Singapore

Supported by:

NUS National University of Singapore | School of Computing

香港教育大學 The Education University of Hong Kong

CoolThink @JC 賽馬會運算思維教育 Inspiring digital creativity 啟發數碼創意

Proceedings of Fifth APSCE International Conference on Computational Thinking and STEM Education Teachers Forum 2021

2nd - 4th June 2021

Singapore

Organized by

Asia-Pacific Society for Computers in Education

Hosted by

National Institute of Education

Nanyang Technological University, Singapore

Publication of Asia Pacific Society for Computers in Education

ISSN 2737-565X

APSCE

Conferences

Preface

The 5th APSCE International Conference on Computational Thinking and STEM Education 2021 (CTE-STEM 2021) is organized by the Asia-Pacific Society for Computers in Education (APSCE). CTE-STEM 2021 is hosted by the National Institute of Education, Nanyang Technological University (NIE/NTU). This conference continues from the success of the previous four international Computational Thinking conferences organised by the Education University of Hong Kong (EdUHK) and JC@Coolthink in Hong Kong. In addition to Computational Thinking, we will be expanding the conference to invite STEM researchers and practitioners to share their findings, processes and outcomes in the context of computing education or computational thinking.

CTE-STEM 2021 is a forum for worldwide sharing of ideas as well as dissemination of findings and outcomes on the implementation of computational thinking and STEM development. The conference will comprise keynote speeches, invited speeches, panel discussions, workshops and paper presentations. All accepted papers will be published in ISSN-coded proceedings.

The International Teachers Forum is organized for teaching practitioners to share their practices in teaching Computational Thinking, Computing and STEM in the classroom. We believe bringing all these would create enriching experiences for educators and researchers to share, learn and innovate approaches to learning through Computational Thinking and STEM education. This year, teachers can participate in Lightning Talks to share ideas about teaching and learning CT.

The Students Forum (BuildingBloCS) is organized by students, for students. It is Singapore's annual Computing education outreach programme. Started back in 2017, it is not only a national computing education outreach programme, but also a platform for leadership development, innovation programme, EVIA (Education & Values In Action) and student-friendly social network. We have been very encouraged by the strong support given by Ministry of Education (Singapore) and many other community and industry partners.

On behalf of APSCE and the Conference Organizing Committee, we would like to express our gratitude towards all speakers, panelists, as well as paper presenters for their contribution to the success of CTE-STEM 2021.

We sincerely hope everyone enjoys and get inspired from CTE-STEM 2021.

With Best Wishes,

Professor LOOI, Chee-Kit

Conference Chair,
CTE-STEM 2021
National Institute of Education
Nanyang Technological University, Singapore

A/P WADHWA, Bimlesh

Conference Co-Chair,
CTE-STEM 2021
National University of Singapore, Singapore

Professor DAGIENÉ, Valentina

Conference Co-Chair,
CTE-STEM 2021
Vilnius University, Lithuania

Main Theme and Sub-themes

"**Computational Thinking and STEM Education**" is the main theme of CTE-STEM 2021 which aims to keep abreast of the latest development of how to facilitate students' computational thinking abilities and STEM development, in the context of computing education or computational thinking. The conference also aims to disseminate findings and outcomes on the implementation of CT development in school and STEM education. There are 19 sub-themes under CTE-STEM 2021, namely:

Computational Thinking and Coding Education in K-12

Computational Thinking and Unplugged Activities in K-12

Computational Thinking and Subject Learning and Teaching in K-12

Computational Thinking and Teacher Development

Computational Thinking and IoT

Computational Thinking and STEM/STEAM Education

Computational Thinking and Data Science

Computational Thinking and Artificial Intelligence Education

Computational Thinking Development in Higher Education

Computational Thinking and Special Education Needs

Computational Thinking and Evaluation

Computational Thinking and Non-formal Learning

Computational Thinking and Psychological Studies

Computational Thinking in Educational Policy

STEM Learning in the Classroom

STEM Activities in Informal Contexts

STEM Education Policies

STEM Pedagogies and Curriculum

STEM Teacher Education and Professional Development

Paper Submissions to CTE-STEM 2021 International Teachers Forum

The Forum received a total of 29 papers by 59 authors from 6 countries/regions (see Table 1).

Table 1: Distribution of Paper Submissions for CTE-STEM 2021 International Teachers Forum

Country/ Region	No. of Authors	Country/Region	No. of Authors
China	15	Indonesia	3
Hong Kong	10	Singapore	18
India	5	Taiwan	8
		Total	59

The Review Panel for the Forum is formed by 18 members worldwide. Each paper with author identification anonymous was reviewed by at least three Review Panel Members. Meta-reviewers then made recommendation on the acceptance of papers based on Review Panel Members' reviews. With the comprehensive review process, 24 accepted papers are presented (see Table 4) at the conference. In addition, there will be 4 short sessions of Techers sharing their CT in classroom experiences (see Table 2).

Table 2: Paper Presented at CTE-STEM 2021 International Teachers Forum

Sub-themes	Number of Papers
Computational Thinking and Coding Education in K-12	3
Computational Thinking and Unplugged Activities in K-12	4
Computational Thinking and Subject Learning and Teaching in K-12	4
Computational Thinking and Teacher Development	1
Computational Thinking and IoT	0
Computational Thinking and STEM/STEAM Education	6
Computational Thinking and Data Science	0
Computational Thinking and Artificial Intelligence Education	1
Computational Thinking Development in Higher Education	0
Computational Thinking and Special Education Needs	0
Computational Thinking and Evaluation	1
Computational Thinking and Non-formal Learning	2
Computational Thinking and Psychological Studies	0
Computational Thinking in Educational Policy	0
STEM Learning in the Classroom	0
STEM Activities in Informal Contexts	0
STEM Education Policies	1
STEM Pedagogies and Curriculum	1
STEM Teacher Education and Professional Development	0
Total	24

Editors

Chee Kit LOOI
Nanyang Technological University

Bimlesh WADHWA
National University of Singapore

Valentina DAGIENĖ
Vilnius University

Beng Keat LIEW
Republic Polytechnic

Peter SEOW
Nanyang Technological University

Ying Hwa KEE
Nanyang Technological University

Long Kai WU
Nanyang Technological University

Hon Wai LEONG
National University of Singapore

Table of Contents

COMPUTATIONAL THINKING AND STEM/STEAM EDUCATION

COMPUTATIONAL THINKING AND ARTIFICIAL INTELLIGENCE

COMPUTATIONAL THINKING AND EVALUATION

COMPUTATIONAL THINKING AND NON-FORMAL LEARNING

STEM EDUCATION POLICIES

STEM PEDAGOGIES AND CURRICULUM

Computational Thinking and Coding Education in K-12

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Teaching Computational Thinking Skills through Debugging with Scratch

Wee Meng Frankie LEOW
Bedok Green Secondary School, Singapore
leow_wee_meng_frankie@moe.edu.sg

ABSTRACT

Debugging is central to students' learning of programming and their development of computational thinking (CT) because when a program does not work as intended, students will need to problem solve by employing CT skills such as breaking the buggy code down into chunks and to devise algorithms to fix the errors. This paper presents the strategies used in teaching students' CT skills through debugging with Scratch, in a typical public co-educational school (hereinafter called "School A") and their implications for teaching and learning.

KEYWORDS

Computational Thinking, Scratch, Debugging, Computer Applications, K-12

1. INTRODUCTION

To prepare our students to be future-ready and able to thrive in an increasing complex and digitalised world, one of the key enablers is to develop Singapore's computational capabilities (Smart Nation, 2014). Hence, the Singapore Ministry of Education (MOE) has introduced the learning of CT and programming into the syllabi for the General Certificate of Education (GCE) N-Level Computer Applications (CPA) subject to strengthen students' digital literacy (MOE, 2019). CT refers to the thought processes in formulating a problem and expressing the solution(s) in ways that an information processing agent (e.g., a computer or human) can effectively implement (Wing, 2017). Secondary students taking CPA learn CT and its related concepts such as algorithmic thinking, abstraction, decomposition and evaluation through engaging in programming activities (writing, testing and debugging codes) to create programs such as animations and games with Scratch, a visual block-based programming language.

2. SCRATCH

Scratch provides a programming environment that offers low floor, high ceiling, wide walls coding experiences for students (Resnick et al., 2009). This allows our CPA students who have minimal prerequisite knowledge to engage with coding, develop CT skills as well as have opportunities to explore and create more complex animations and games based on their interests. With its block-based and visual interface, Scratch allows students to write a program by selecting graphical colored-coded blocks of instructions and connecting them vertically to form series of connected blocks called scripts. Figure 1 shows the script of an animation created using Scratch.

3. DEBUGGING

Due to errors or 'bugs' (e.g., logic errors) present in the algorithms and code in the script, a program may not always work exactly as intended. For example, Figure 1 should show the script that is meant to draw a set of ten squares, one inside the other. However, due to off-by-one error in the repeat loop, an unintended output is generated.

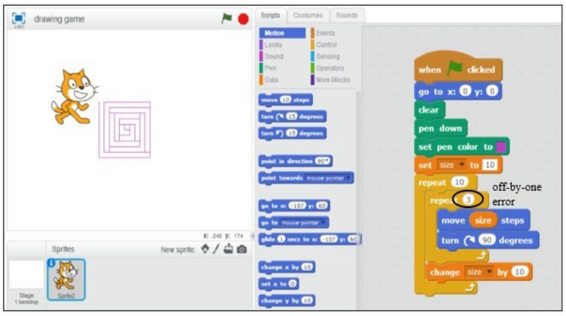


Figure 1. Example of an off-by-one error in the script of an animation created using Scratch

Hence, students need to perform debugging: the process of finding and correcting these errors (Berry, 2017; Kazimoglu, et al., 2012) so that the code compiles successfully and executes to generate the expected results. As students test and debug buggy code(s) in the script using debugging strategies, they will use CT skills such as logical reasoning and pattern recognition when they predict what will happen when they go through their algorithms and code, and to explain their thinking; decomposition and abstraction when they break the scripts down into component chunks and filter out the redundant detail to find and correct the error(s). Thus, debugging is central in developing students' CT skills (Berry, 2015; Wing, 2017).

4. DEVELOPING CT SKILLS THROUGH DEBUGGING STRATEGIES

At school A, secondary two CPA students were first taught explicitly the following strategies, rubber duck debugging and wolf fence debugging, followed by the steps in the debugging procedure (DP) to debug their own buggy codes. Thereafter, students were given debugging activities to complete so as to further enhance their debugging skills.

4.1. Rubber Duck Debugging

A strategy that students learned in order to find the cause of the problem is to explain it to someone else, like a rubber duck (Hunt & Thomas, 2000). When students' programs go wrong, they would be given a rubber duck for them to explain to the ducks what their program should do, and what it actually does. Starting from the first block of the script, they will read and explain to the duck, line by line, what the code is supposed to do. In doing so, students are verbalising what the problem is, externalising their thoughts, and paying closer attention to what is really present in the code, until the error(s) is detected and fixed.

4.2. Wolf Fence Debugging

If the script is long and complex, students are taught to work out which section of the code has the error(s) by breaking the entire code down into chunks to check and test the code (Gauss, 1982). For each chunk, the cycle is repeated for that part of the code and students will eliminate the areas repeatedly until the block(s) causing the problem has been found (see Figure 2).

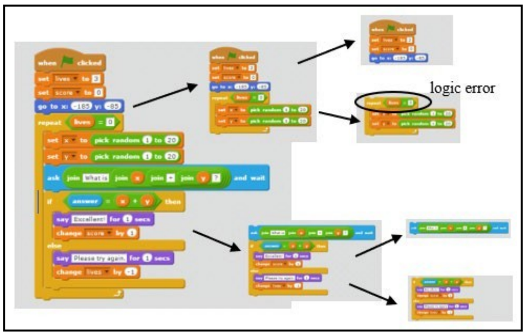


Figure 2. Example of wolf fence debugging

4.3. Debugging Procedure

Subsequently, students are taught a DP shown in Figure 3 for them to follow to debug their buggy code, based on their observation of the program's output. The procedure will require students to use their previously learned debugging strategies so as to debug successfully.

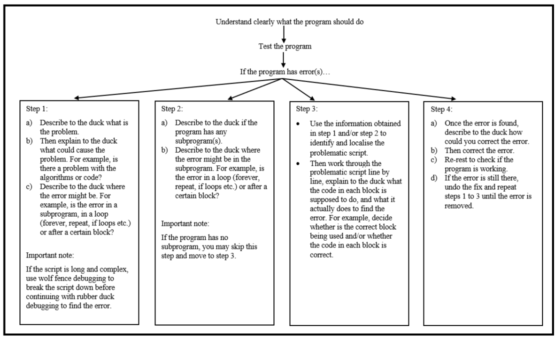


Figure 3. Chart showing steps in debugging procedure

4.4. Debugging Activities

In addition, students will also complete two debugging activities to further improve their debugging skills. The first activity involves students completing debugging exercises independently. Students will be given worksheets and buggy codes for them to identify and correct the errors present (see Figure 4).

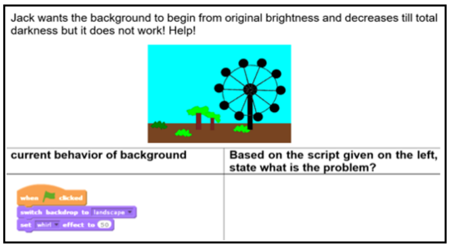


Figure 4. Example of a debugging exercise

The second activity requires students to create their own debugging challenge for others to solve by sabotaging one another's code (O'Donohoe, 2013). One student will swap seats with his/her partner to introduce a fixed number of errors before swapping back as the original programmer debug these new errors. The errors introduced by the saboteur can focus on a specific CT concept, the algorithms or Scratch blocks of the code etc.

5. REFLECTIONS

Being novice programmers, the main problems students faced are the difficulties with which they are unable to locate the error(s) and making counterproductive changes to the code while debugging such as introducing new error(s) during debugging. During the debugging activities, I observed that many students are now able to apply CT skills as they debug, using a two-level analysis to code recognition and error localisation. For novices, they first break down a buggy code into different chunks (decomposition), analyse the sequence of the blocks for correctness (algorithmic thinking), and identify the focal points for granular analysis after removing redundant details (abstraction). Thereafter, the novices articulate aloud what each block should do, and what it actually does, (logical reasoning and pattern recognition) as they examine the code in each chunk at the micro level. After debugging, they will retest the code again (evaluation). The cycle is repeated if error(s) still persist. And as novices gain familiarity with the functions of the different blocks and common errors through the debugging activities, they gain expertise in debugging.

6. IMPLICATIONS AND CONCLUSION

Errors in programs are varied and teachers' observations suggest that debugging them required an eclectic mix of the CT skills. For novices and experts, CT skills can be taught through the debugging strategies with Scratch. In the process, students will develop and enhance their CT skills.

7. REFERENCES

Berry, M. (2015). *QuickStart computing – A CPD toolkit for primary teachers*. Swindon: BCS.

Berry, M. (2017). *QuickStart computing: Subject Knowledge Enhancement for secondary teachers*. Swindon: BCS.

Gauss, E. J. (1982). The "Wolf Fence" algorithm for debugging. *Communications of the ACM*, *25*(11), 780.

Hunt, A., & Thomas, D. (2000). *The pragmatic programmer: From journeyman to master*. Boston: Addison-Wesley.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, *9*, 522 – 531.

MOE. (2019). *N-Level computer applications syllabus*. Retrieved December 20, 2020, from https://www.moe.gov.sg/docs/default-source/document/education/syllabuses/sciences/files/2019-computer-applications-syllabus.pdf

O'Donohoe, A. (2013, November 23). Sabotage: Teach debugging by stealth. *Teachcomputing*. https://teachcomputing.wordpress.com/2013/11/23/sabotage-teach-debugging-by-stealth/

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60-67.

Smart Nation. (2014). *Why Smart Nation*. Retrieved December 20, 2020, from https://www.smartnation.sg/about-smart-nation

Wing, J.M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7-14.

面向计算思维能力发展的思维型编程教学实践：内涵阐释与框架重构

徐恩伟

新疆师范大学教育科学学院，中国·新疆乌鲁木齐

244924608@qq.com

摘要

研究依据计算思维的本质内涵和思维型教学理论构建了兼具计算思维"计算特征和思维属性"的"CTAD—TPTM"思维型编程教学结构模型和具身化实施的"A—IPO—D"教学实践路径，期望为思维型编程教学的设计与实施提供实践性参考。

关键词

计算思维; 思维型编程; 结构模型; 实践路径

1. 计算思维的内涵解读与实践审视

计算思维最早由麻省理工学院西蒙·派珀特教授于 1980 年首次提出并将其阐述为"儿童在通过计算机学习时所训练与培养的思维技能"(Papert,1980)；随后周以真教授于 2006 年从计算机科学的视角对其重新进行界定，认为"计算思维是运用计算机科学的思维方式和基础概念进行问题解答、系统设计，像计算机科学家一样思考问题、理解问题、解决问题等一系列涵盖计算机科学的思维活动"(Wing J M,2006)。2011 年，ISTE 和 CSTA 共同提出了 K-12 教育中计算思维培养的操作性定义，即"计算思维是一种问题解决的思维过程，包含'借助工具分析问题、有逻辑性的处理数据、利用算法自动化解决问题，以及解决问题过程的迁移运用'等步骤"(ISTE and CSTA,2007)；Brennan 等人在此操作性定义基础之上基于 Scratch 儿童编程环境构建了包含"计算概念、计算实践、计算观念"的计算思维三维框架，成为广泛被接受的计算思维教学的支撑理论及实践指导(Brennan,2012)；John Woollard 等人在概述前人研究的基础之上提炼出"抽象、分解、算法思维、概括和评价"(Cynthia Selby&John Woollard,2013) 五个计算思维的核心要素，不仅成为问题解决的五种科学计算方法，而且成为教学实践中检验计算思维培养效果的重要评价指标。

上述关于计算思维的定义虽然迥异，但其内涵聚焦于两个方面：一是问题解决，二是思维活动，即计算思维不仅是一种利用计算机工具进行问题解决的能力而且是一种问题解决过程中内在思维活动的表现，兼具问题解决的"计算特征"和思维活动的"思维属性"。但由于计算思维起源于计算机领域，教师对其认知和理解仅停留于"计算"层面，在编程教学实践中多专注于其"计算"特征而

忽略了"思维"属性，致使智能时代计算思维的价值意义与教学实践的培养效果存在差异，突出表现在两个方面：一是课堂教学内容层面侧重编程工具的学习而非思维能力的发展，虽然图形化编程、Python 文本编程等编程教育课程陆续开展，但在实际的教学过程中过于追求编程技术的新鲜感，致使学生以"尝试一次"的技术工具学习为主，无法达成对问题情境的深刻认识和解决实际问题的思维能力；二是教学模式层面教师预设解题路径而非指向思维的探究式意义建构。教学中虽然采用了任务驱动、基于问题或基于项目等若干教学策略，但"教师为追求问题解决的课堂效率往往预设学习路径和任务的操作步骤，并要求学生根据操作步骤依次完成学习任务，致使编程教学成为模仿和重复操作的无思维学习"(顾坚,2018)。

面对计算思维培养效果甚微的实践困境，如何在编程课堂教学中兼顾计算思维的"问题解决的计算特征和思维活动的思维属性"，有效促进学生计算思维能力的发展，是基于核心素养课程改革迫切需要解决的一个问题。

2. "CTAD—TPTM"结构模型的构建

2.1 思维型教学理论

思维型教学是聚焦于培养学生思维能力发展的教学，当前有两种基本的实践形式："独立课程的思维型教学和融合课程的思维型教学"(赵国庆,2013)。前者是在学校里单独开设思维型课程，通过专门的老师开展课堂教学活动以提高学生的思维能力，其教学目标是要让学生"知道如何思维"；后者是把思维能力的培养与学科教学紧密融合在一起，通过教师聚焦思维能力的课程设计与活动实施，使学生在获取学科知识的同时发展思维能力，其教学目标是要让学生"迁移创新"。然而，所有知识的学习都涉及到思维，并不存在无思维的教学，因此在学科教学中发展学生的思维能力被认为是培养学生思维能力的有效实践路径(McGuinness,2007)。林崇德等人把思维活动作为课堂教学中师生活动的核心，并依据"聚焦思维结构的三棱模型"提出了思维型教学的四大基本原则和四个基本环节，即"认知冲突、自主建构、自我监控、应用迁移"原则和与之对应的"教学导入、教学过程、教学反思、应用迁移"环节(林崇德和胡卫平,2010)。

思维型教学理论强调课堂活动的核心是对学生思维能力的培养以促进迁移创新，其"四大基本原则"和"四个基本环节"为研究思维型编程教学提供了理论指导和操作指南，然而其并未指出如何将计算思维与编程教学进行融合以及通过编程教学发展学生的哪些思维素养。

2.2《新课标》关于"计算思维"定义所蕴含的思维素养

我国《普通高中信息技术课程标准（2017 版）》指出："计算思维是指个体运用计算机科学领域的思想方法，在形成问题解决方案的过程中产生的一系列思维活动。具备计算思维的学生能够采用计算机科学领域的思想方法界

定问题、抽象特征、建立结构模型、合理组织数据；通过判断、分析与综合各种信息资源，运用合理的算法形成解决问题的方案；总结利用计算机解决问题的过程与方法，并迁移到与之相关的其他问题解决中”。(中华人民共和国教育部.普通高中信息技术课程标准,2018)

其中“界定问题、抽象特征、建立结构模型”蕴含“问题思维”的培养，智能时代将会面临各种“不确定性”的问题，是否能够分析情境中蕴含的问题并进行界定、对其特征进行抽象化诠释进而建立问题特征的结构模型，需要学生具有积极主动思考的问题思维能力。“合理组织数据、运用合理的算法形成解决问题的方案” 蕴含“批判性思维、算法思维”的培养，其中“合理”即批判性思维，需要学生在分析评估的基础上辨证性地提出质疑、批判性的选择以监督问题解决沿着正确的方向行进，“算法”即算法思维，是一系列定义良好的待执行任务的逻辑步骤，步骤的排列顺序是算法思维的具象化体现。“判断、分析与综合各种信息资源”蕴含“协作思维、批判性思维”的培养，学习是在师生交互与协作的活动过程中达成的，在协作中不仅需要学生通过批判性思维对各种信息资源进行批判性的质疑选择和迁移应用，而且协作本身蕴含情感交互，“协作思维”有助于共同体学习情感的提升。“迁移到与之相关的其他问题解决中”蕴含“创新思维”的培养，迁移所学知识和能力以创造性地解决问题是创新思维的体现、是编程教学的终极指向、亦是适应智能时代问题解决的关键能力。

综上分析可知，《新课标》中对计算思维概念的界定蕴含了“问题思维、批判性思维、协作思维、算法思维、创新思维”等五种指向计算思维能力发展的具象化思维素养。

2.3 “CTAD—TPTM” 结构模型及其诠释

本研究依据计算思维的本质内涵和思维型教学理论构建了兼具“计算特征和思维属性”的指向计算思维能力发展的思维型编程教学结构模型（英文首字母缩写为“CTAD—TPTM”）（图 1）。

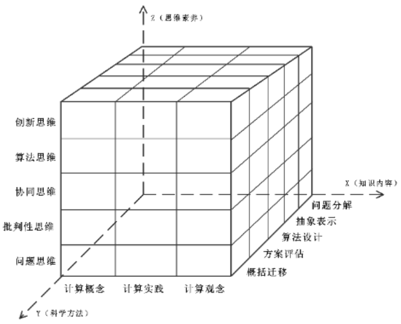


图 1　CTAD--TPTM 结构模型

该教学结构模型体现出三个特点：

第一，模型共有三个维度，x 轴表示计算思维的知识内容，包含“计算概念、计算实践、计算观念”3 个要素，计算概念是指编程教学的核心概念，如序列、变量、数据、循环等概念知识；计算实践是指运用计算概念实施的问题解决过程；计算观念是指学习者在计算实践过程中形成的价值倾向；y 轴表示计算思维的科学方法，包含“问题分解、抽象表示、算法设计、方案评估、概括迁移”5 个

要素；z 轴表示计算思维聚焦的思维素养，包含“问题思维、批判性思维、协作思维、算法思维和创新思维”5 个要素。每个维度的要素相互组合，可形成 75(3*5*5)个结构单元，每个结构单元作为思维性编程教学所指向的计算思维能力发展状态。

第二，模型将计算思维的计算特征和思维属性蕴含其中：所谓计算特征即指在“计算概念、计算实践和计算观念”的三维框架中使用“问题分解、抽象表示、算法设计、方案评估、概括迁移”等五种常见的计算机科学领域的方法对问题进行求解；所谓思维属性即指在问题求解过程中聚焦“问题思维、批判性思维、协作思维、算法思维、创新思维”等创新的思维活动。

第三，模型以计算思维的知识内容为载体，通过计算机科学领域的方法聚焦于思维素养能力的达成，既符合知识、方法、能力之间的关系论述，同时揭示了思维型编程教学中计算思维能力的发展路径：一方面计算思维能力是由其知识内容、计算方法、思维目标共同构成的一个有机整体，在教学实践中不可忽视任何一个维度；另一方面，教学实践中需要将计算思维所聚焦的思维素养作为教学的目标指引，引导学生不断探索并完善模型的结构单元以促进其计算思维能力的良善发展。

3. 具身化实施：“A—IPO—D”教学实践路径

“CTAD—TPTM”教学结构模型虽然从理论上诠释了计算思维的知识内容、计算方法和聚焦思维素养的有机统一，为教学实践中计算思维能力的发展指明了培养方向，但其抽象的理论解释无法为编程教学实践提供可操作性的方法指南。在编程设计中最常见且最基本的方法是 IPO 法（“输入—处理—输出”的英文缩写），但其只是一般的程序设计方法并未指出如何在教学中培养计算思维。因此，研究依据“CTAD—TPTM”教学结构模型和编程设计的 IPO 法，构建了“CTAD—TPTM”具身化实施的“A—IPO—D”（Analysis situation --Input--Processing--Output--Display Communication）教学实践路径（图 2）。

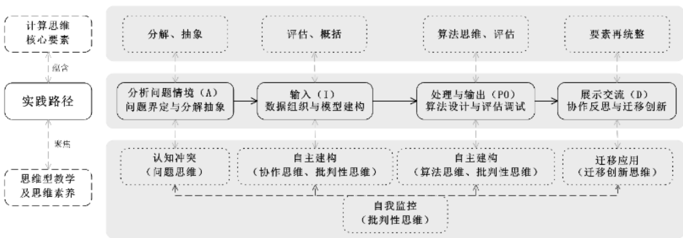


图 2 A—IPO—D 教学实践路径

“A—IPO—D”实践路径包含四个教学环节，每个环节都有相对应的计算思维核心要素和思维型教学原则及其聚焦的思维素养，具体实施要点如下：

（1）分析问题情境（A）：问题界定与分解抽象

分析情境中的问题是编写程序的开始，问题情境是引发学生认知冲突、激活思维的关键。在具体的问题情境中学生通过新旧知识与经验的互联，进行问题界定、分解细化并且能够对其特征进行抽象化概括，这是形成问题解决的关键。此环节通过问题情境激发学生的认知冲突、引发学生对问题本质的思考，即学生能否使用自己的语言解

释情境中需要解决的问题？如何从情境中抽象出问题的基本特征以及涉及到的学科核心概念知识？是否能够以及如何将问题分解成较小的组成单元？通过这种思考引发学生对计算概念的理解，培养学生的问题思维，达成“问题特征抽象和问题分解”的计算思维要素。

（2）输入（I）：数据组织与模型建构

输入是一个程序运行的开始，其方式一般包括：文件输入、网络输入、用户键盘输入、数据库输入、程序内部参数输入等，对应的数据类型有数字型、列表、集合、元组、字典等，即输入主要围绕数据的获取、组织以建构数据的组织模型。数据作为信息技术学科的核心概念，对其准确的理解和组织是培养计算思维的关键。此环节通过抽象出的问题特征引发学生对数据的思考，即学生能否清晰地说出问题解决中需要哪些数据？这些数据的基本类型是什么以及如何获取？能否通过流程图的形式建构数据的组织模型？如何综合协作小组成员的知识成果批判性的思考数据选择与组织模型建构的合理性？通过这种思考引发学生对计算实践的理解，在协作学习中培养学生的批判性思维和协作思维，达成“评估和概括”的计算思维要素。

（3）处理（P）与输出（O）：算法设计与评估调试

程序运行的逻辑需要对输入的数据进行处理与输出，亦即算法设计和评估调试。算法是数据组织模型的具体化实施，是计算思维的具体实现方案，输出是展示算法设计运算成果的方式。然而任何程序功能的实现都不是一蹴而就的，需要在输出过程中不断地对出现的问题进行调试评估。此环节通过数据的组织模型引发学生对算法与编程工具的思考，即学生能否找到或设计出解决问题的算法逻辑？能否使用序列、变量、循环、条件语句等计算概念列出编程问题解决的基本操作步骤？如何选择合适的编程工具将列出的编程步骤组合成计算机识别的行为序列以形成可执行的程序？是否能够根据程序执行的过程和结果调试出现的错误以及评估该算法程序解决问题的合理性？通过这种思考引发学生对计算实践的再理解，培养学生的算法思维和批判性思维，达成“算法思维和评估”的计算思维要素。

（4）展示交流（D）：协作反思与迁移创新

展示交流是让学生表达自己对情境问题解决与编程实现等有关知识与方法理解和感悟的反思过程，可依据教学任务和学时安排设计协作反思与迁移创新两个环节。协作反思是以协作小组的形式对情境任务的问题解决方案进行评估与概括，以内化学生利用所学知识与技能进行迁移创新的能力；迁移创新是基于任务的问题再解决，包含基础任务、能力任务和创生任务三个难度层次，从而将内化的迁移创新能力再现出来。此环节通过协作反思与迁移创新引发学生对编程问题解决过程与方法的总结与反思，即学生能否解释编程问题解决方案中涉及的核心概念与技巧方法？是否能够反思该编程方案解决情境问

题的满意度如何以及如何改进？是否能够运用已掌握的知识和能力独立解决相似情境下的基础任务、能力任务和创生任务？通过这种思考引发学生对计算观念的理解和计算实践的再实践，培养学生的迁移创新思维，达成计算思维各要素的再统整性训练。

4. 结语

因缺乏对计算思维概念本质内涵的批判性理解，教师在编程教学实践中多专注于其“计算”特征而忽略了“思维”属性，致使智能时代计算思维的价值意义与教学实践的培养效果存在差异。面对编程教学培养计算思维效果甚微的实践困境，研究在教学实践的基础上依据计算思维的本质内涵和思维型教学理论构建了兼具计算思维“问题解决的计算特征和思维活动的思维属性”的“CTAD—TPTM”思维型编程教学结构模型和其具身化实施的“A—IPO—D”教学实践路径，不仅能够为学术研究者和教学实践者研究计算思维能力的发展提供新的研究视角，而且能够为思维型编程教学的设计与实施提供实践性参考，进而从整体上促进学生计算思维能力的发展，促使其成为智能时代问题的解决者和创造者。

5. 参考文献

顾坚(2018).计算思维在初中信息课程中的实践研究.上海:上海师范大学

赵国庆(2013).思维教学研究百年回顾.现代远程教育研究,2013(06):45.

林崇德和胡卫平.思维型课堂教学的理论与实践.北京师范大学学报(社会科学版),2010(01):29-34.

中华人民共和国教育部(2018).普通高中信息技术课程标准(2017 年版).北京:人民教育出版社.

Papert(1980). Mindstorms: Children, computers, and powerful ideas. New York:Basic Books

Wing J M(2006). Computational thinking. Communications of the ACM, 2006(3):33-35.

ISTE and CSTA(2007). Operational definition of computational thinking for K-12 education. Retrieved 4 28, 2018, from <http://www.iste.org/docs/ct-documents / computational-thinking-operational-definition-flyer.pdf>

Brennan(2012). New frameworks for studying and assessing the development of computational thinking. Vancouver: AERA,2012.39-48.

Cynthia Selby&John Woollard(2013). Computational Thinking: The Developing Definition. Special Interest Group on Computer Science Education -2013.

McGuinness(2007).Building Thinking Skills in Thinking Classrooms: ACTS in Northern Ireland. The 13th International Conference on Thinking Norrköping, Sweden :Linköping University Electronic Press

Thinking Programming Teaching Practice Oriented Computational Thinking Ability Development: Connotation Interpretation and Framework Reconstruction

Xu Enwei

School of Educational Science, Xinjiang Normal University, Urumqi, Xinjiang, China
244924608@qq.com

ABSTRACT

Based on the essential connotation of computational thinking and thinking-based teaching theory, the study constructed a "CTAD-TPTM" thinking programming teaching structure model with both "computational characteristics and thinking attributes" of computational thinking and an embodied implementation of "A-IPO-D" teaching practice path is expected to provide practical reference for the design and implementation of thinking programming teaching.

KEYWORDS

Computational Thinking; Thinking Programming Teaching; Structural Model; Practice Path

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021*. Singapore: National Institute of Education.

透過 Scratch 培養學生運算思維之教學實踐

楊詠盈 [1]，冼文標 [2]*
[1,2] 鳳溪第一小學，香港
wingying@fk1ps.edu.hk，mpsin@fk1ps.edu.hk

摘要

近年來不少地方都推廣「運算思維」的教學，學生需 應用這種思維技能進行解難。編程教學着重學生思維 訓練，然而這非單一在資訊科技課內學習。本文將分 享校本***跨學科教學***的經驗，將編程教學融入數學科之內，學生能寓編程於數學學習中。學生的學習目標明確，活動多元，學習興趣也提高了。跨學科協作更突 顯運算思維在日常生活的應用，更容易培養學生的思 維訓練。

關鍵字

運算思維；跨學科編程；八個方向

1. 前言

科技發展一日千里，學與教也與時並進。在 2015 年香港政府首次提及推動 STEM 教育，香港教育局(2016)在《推動 STEM 教育發揮創意潛能》的課程文件中建議在小學引入編程，以發展學生的計算思維，鼓勵資訊 科技科教師與各位教師共同合作，提供機會讓學生通 過適當設計的學習活動，學習和應用編程的技能(頁 13)。運算思維應該是所有人共通具備的能力，善用這 項能力可以增加解決問題的能力，培養邏輯思考、系 統化思考等運算(邵雲龍，2019)。為裝備學生與世界接軌，香港教育局(2020)亦在運算思維教學上亦開始推出《計算思維— 編程教育小學課程補充文件》，提倡了第二學習階段(小四至小六)的編程教育。期望學生在編 程教學中以實作經驗，建立解難的信心，透過協作及 重覆的測試來解決問題。近年於小學課程中較常見的 就是 Scratch，學生透過 Scratch 編寫程式，也可以在社群內分享及再創造。本校在小學三年級資訊科技科課 程內已開始加入編程教學，期望藉着編程發展學生的 運算思維發展。運算思維不是單一領域的學習目標，而是學生未來應具備的素養。故此本校也鼓勵跨科合作， 把思維學習與其他知識點連結， 使學習是相承而非割裂。本文將分享一個 Scratch 編程的跨科應用例子，以 Scratch 作為運算思維教學的推動，將學習點貫通。

圖 1： 運算思維

2. 運算思維(Computational Thinking,CT)

運算思維強調問題解決過程中，利用電腦科學提高解 決效率的能力，也是一種心智的工具，更是每個人應 具備或最好具備的能力 (黃蕙蘭等，2020) 。 也就是說運算思維是一種利用電腦的邏輯來解決問題的思維， 就是一種能夠將問題從抽象到具體的能力。Google for Education 提出培養運算思維的四個面向(圖 1)，分別是拆解(Decomposition)、模式識別(Pattern Recognition)、歸納與抽象化(Pattern Generalization and Abstraction)與演算法(Algorithm)。在日常生活事件中，人們在解決問題的過程也可運用「運算思維」的邏輯去思考，把問 題分拆，再經歷過程去解難。

3. Scratch 跨科學習專案

按照數學教育學習領域課程指引補充文件(2017)提出小學四年級的課程中學習方向和位置(「八個方向」)。學生在此學習階段能認識的方向包括東、南、西、北、 東南、東北、西南、西北。在基本教學流程中，教師 會藉課本內的題目教授學生相對位置及用方向描述及 規劃路線。方向的題目類型單一，反覆操練會使學生 的學習動機降低。另外，在資訊科技科的課程，四年 級學生正學習編寫迷宮。有見及此，兩科教師合作，配合資訊科技科以 Scratch 迷宮

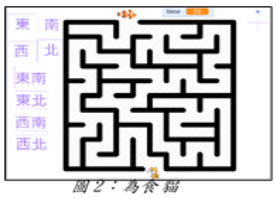


圖 2：為食貓

教學內加入八個方位學習，一方面是可讓學生透過自 擬學習方向的情景介面提升學習動機，另一方面是可 深化方向教學，以程式碼協助設計遊戲。

表 1 陳述了以 Scratch 學習八個方向之運算思維概念，圖 2 為教師範例 – 為食貓。

表 1：運算思維於方向學習

運算思維	***內容***
拆解	數學：分析地圖
模式識別	數學：根據方向板的位置，判別各個方位
歸納與抽象化	數學：規劃路線
演算法	資訊：加入方向鍵配合相應的程式碼

4. 編程相關的教學內容

學生的先備知識是八個方向，教師再將極座標系統簡化地向學生提出即，故對於 方向板時，角色面朝北則是面朝 0°，東方則是面向北方時左轉一個直角，即 90°，東北方

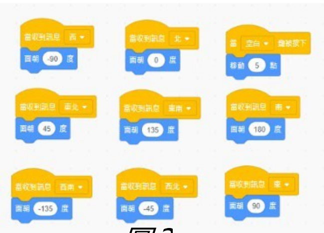


圖3

則是半個直角，即面朝 45°，如此類推(圖 3)。在這個教學中，是學習簡單方向系統的延伸，對於八個方向 之間的關係深化。在教授了上述基本教學後，學生可 以自行按自己放置的方向板及地圖作出修改。程式碼 會因方向板的方向而需改變。而在原本的迷宮程式碼 上，加入「方向」按鈕， 用方向指令角色移動的位置。由於在編程後學生會進行測試，這個過程需要學 生對八個方向完全掌握才可走出迷宮。

5. 學習成果

編寫迷宮是學生已掌握的知識， 所以在課堂上再新增數學概念，遊戲可設計得更多元，在編程的同時也可深化數學內容。圖 4 至圖 6 是部份學生的作品，圖 4 是屬基本學生能完成的作品。在教師的範例中主角只用了四個方向，小貓就能吃魚。所以學生需要再創造，把迷宮加入其他要行

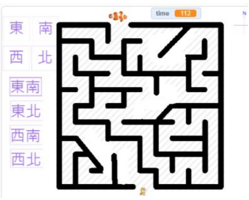


圖 4：學生作品 1

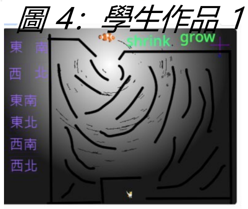


圖 5：學生作品 2

走的方向，使小貓需用齊八個方向才能吃魚。而圖 5，學生把方向板的方向調了位置，所以方向編碼也重組了，而圖學生還自創了角色放大及縮小鍵，令角色可在一些難通過的地方用縮小鍵。而圖 6 的學生則將主題改編成海洋世界。

圖 6：學生作品 3

6. 反思及結語

在這次跨科學習專題中，令學生對八個方向的課題更 深刻。每一次測試都是練習，比起普通在教科書內的 題目更多元。在編程方面，學生能掌握基本迷宮的設 計，應用運算思維去創作，這次機會令學生了解知識 是可以共融的。無論對數學或是編程，更引起學生學 習興趣，這比海量式練習更具意義。

這次是兩科教師跨科課題合作在編程教學中，學生主 動學習的結果卻令教師滿意。有部份學生明白運算思 維原理後，更編寫出幾個不同版本的方向迷宮，可見 學生充滿創意，這些創意就是學生運算思維能力的顯 證。跨科協作需要教師互相合作，讓同工對不同學習 領域的了解，各科學習才可考慮借助編程深化教學目 標。資訊科技的實用性更可彰顯。

7. 參考文獻

課程發展議會(2016)。推動 STEM 教育發揮創意潛能報告。擷取自網頁 https://www.edb.gov.hk/attachment/tc/curriculum-development/renewal/STEM_Education_Report_Chi_20170303.pdf

課程發展議會(2020)。計算思維– 編程教育小學課程補充文件。擷取自網頁 https://www.edb.gov.hk/attachment/tc/curriculum-development/kla/technology-edu/curriculum-doc/CT_Supplement_Chi%20_2020.pdf

課程發展議會(2017)。數學教育學習領域課程指引補充文件。擷取自網頁 https://www.edb.gov.hk/attachment/tc/curriculum-development/kla/ma/curr/pmc2017_tc.pdf

黃蕙蘭、黃思華、黃健哲(2020)。國小一年級學童實施不插電運算思維課。臺灣教育雙月刊，772(1)，P.59-70

邵雲龍(2019)。視覺化程式融入運算思維之教材發展與評估。先進工程學刊， 14(2)， P103 – 110。

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

The Teaching Practice of Cultivating Students' Computational Thinking through Scratch

Wing Ying YEUNG[1], Man Piu SIN[2*]
[1,2]Fung Kai No.1 Primary School, Hong Kong
wingying@fk1ps.edu.hk, mpsin@fk1ps.edu.hk

ABSTRACT

In recent years, many countries have promoted computational thinking. Students need to apply this thinking skill to solve problems. The teaching of Programming focuses on students' thinking training, but this is not just learning in the IT lesson. This article will share the experience of school-based interdisciplinary teaching, integrate programming teaching into mathematics, and students can embed programming in mathematics learning. Students have clear learning goals. Those diverse activities can help increasing students’ interest in learning both mathematics and programing. Cross-disciplinary collaboration highlights the application of computational thinking in daily life and makes it easier to cultivate students' thinking training.

KEYWORDS

computational thinking, interdisciplinary, directions

Computational Thinking and Unplugged Activities in K-12

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

SWOT Analysis and Strategy of Unplugged Activities to Localize STEM Courses in Rural Schools

Jiashuo CHANG[1], Shuo GUO[2*]
[1,2] School of Education, Shaanxi Normal University, China
changchangjiashuo@163.com, 2797251338@qq.com

ABSTRACT

Computational thinking is an essential quality for digital citizens in the information society. How to cultivate students' computational thinking is a problem that researchers pay close attention to. Under the guidance of "promoting the integrated development of urban and rural compulsory education" and "education equity", the cultivation of computing thinking should not only stay at the urban level where computer programming education is carried out in full swing, but also extend to the rural areas. However, facing the backwardness of rural economic development and the restriction of hardware environment, we should think about whether computing thinking must be cultivated in the computer environment? Unplugged activities help learners explore and understand the subtle ideas of solving problems autonomously and creatively through real life situations and projects without computer support, thus cultivating students' computational thinking. Based on the analysis of SWOT, this paper carried out unplugged activities in rural areas to help the development of computing thinking. Unplugged activities are undoubtedly an economical and affordable choice for areas with backward economic conditions and lack of high-quality teaching resources to promote the development of computing thinking.

KEYWORDS

Unplugged Activity, Rural STEM Education, SWOT Analysis, Unplugged Computer Science, Computational Thinking

1. INTRODUCTION

Computational thinking is a problem-solving thought process that clearly and abstractly expresses problems and solutions in a way that information processing agents can effectively perform. What is it to cultivate students' computational thinking? Is it just for students to become computer experts? In fact, this is not the case. The computational thinking we cultivate should refer to a series of thinking activities produced by individuals in the process of forming solutions to problems by using the thinking methods of the computer field (Ministry of Education, 2017). The students in K12 stage are in the critical period for their thinking and ability cultivation and development. It is extremely important to implement and promote computing thinking education in this stage. That is to say, computational thinking is a basic skill that all digital citizens in the information society need to master, and it is also one of the essential core qualities of learners in K12 stage. STEM education, as a fertile ground for cultivating students' computational thinking, has been playing an important role. STEM education is an interdisciplinary education that integrates science, technology, engineering and mathematics. The four disciplines are organically integrated, with real problems or goals as the orientation, and students' creativity, problem-solving ability and interdisciplinary awareness are cultivated in practice. In recent years, the Chinese government has vigorously advocated STEM education, maker education and other new education modes, making computer programming education, as the main force, widely carried out in K12 stage. However, according to the relevant research at home and abroad, the current theoretical research and practical exploration on computational thinking and STEM education based on computer programming are mainly concentrated in urban primary and secondary schools, while few are involved in rural primary and secondary schools. According to the national primary school enrollment data from 2018 to 2019, the number of rural primary school students accounts for about 24 percent of the total number of primary school students in China. The cultivation of computational thinking of students in rural areas cannot be ignored. The report to the 19th National Congress of the Communist Party of China (Xi Jinping, 2017) pointed out: "We should attach great importance to compulsory education, promote the integrated development of urban and rural compulsory education, and strive to ensure that every child can enjoy fair and quality education." Therefore, the cultivation of computational thinking should not only stay at the urban level where computer programming education is carried out in full swing, but also extend to the rural areas.

However, there are some differences between urban primary and secondary schools and rural primary and secondary schools, such as teachers, school philosophy, hardware and software facilities, students' ability basis, etc. Therefore, the existing computer programming teaching mode and teaching means cannot be directly copied. Primary and secondary schools in rural areas are backward in economic conditions and lack of high-quality teaching resources, which makes it impossible to achieve complete computer equipment for programming education. Given this lack of hardware, we tried to figure out if we could cultivate computational thinking in a non-computer environment.

2. UNPLUGGED ACTIVITY

The cultivation of computational thinking in rural areas deserves more attention. However, there are some differences between urban primary and secondary schools

This paper was supported by the Fundamental Research Funds For the Central Universities, Innovative team project for graduate students of Shaanxi Normal University (Project No.TD2020009Y).

and rural primary and secondary schools, such as the level of teachers, school philosophy, hardware and software facilities, students' ability basis, etc. Therefore, the current existing computer programming teaching mode and teaching means cannot be directly copied. Primary and secondary schools in rural areas are backward in economic conditions and lack of high-quality teaching resources, which makes it impossible to achieve complete computer equipment for programming education. Given this lack of hardware, we tried to figure out if we could cultivate computational thinking in a non-computer environment. Unplugged activity offers us a practical way.

In 1999, Tim Bell, Ian H. Written and Mike Fellows in New Zealand proposed the "Unplugged Computer Science" teaching concept (Bell, T., Witten, I.H., Fellows, M., 1999), which aims to learn computer science concepts without computers by role-playing or using physical objects such as paper, pens and cards. Unplugged computer science is suitable for students from different countries and with different levels of knowledge. Unplugged computer science helps learners to explore and understand the subtle ideas of computing, operation and problem solving through "learning by play, learning by doing" without computer support, stimulating students' interest in learning and cultivating students' computational thinking.

Based on the concept of unplugged computer science, Tim Bell later proposed nine principles for unplugged activities (Bell, T., 2019). Now let's use these nine principles to understand what it means to be unplugged. The first is the “activity” in the term "unplugged activity," which means that the activity is usually large-scale, not just a one-person learning process, but a team effort. At the same time, the activity should be interesting and engaging, it can lead to the content of the activity with a story, so that learners can find the answer independently in the play, rather than just busy operation. The second is the “unplugged” of the word "unplugged activity", which means the removal of computers from the teaching of computational thinking without deviating from the teaching objectives of computational thinking. Because computational thinking is not necessarily using computers to solve problems, but using ideas and methods in computer science to solve practical problems (Dou,Y.,2015). Computer programming may become the bottleneck for students in K12 stage, and learning programming directly on the computer will increase the cognitive load of learners. But in unplugged activities, learners are away from the computer, and they are able to think about problems in real situations, rather than just focusing on the computer itself.

3. SWOT ANALYSIS OF UNPLUGGED ACTIVITY IN RURAL SCHOOLS

Due to the limitations of teachers and school conditions in rural schools, school running philosophy and teaching methods are backward, and input in information construction is relatively weak. As a result, students' overall basic ability is poor, and information literacy and information awareness are shallow. In the survey, almost all students had never taken a STEM course or a course related to the development of computational thinking, and even information technology courses were not guaranteed to be carried out smoothly. Through communication with some of the students, they showed great interest in the courses of information technology and scientific exploration. On the whole, students are weak in the application of information technology, lack the ability to independently solve interdisciplinary problems, lack the sense of cooperation and innovation, but they show initiative and enthusiasm in learning STEM courses. The development and implementation of rural STEM education are affected by political foundation, economic level, education level, people's educational needs and understanding and other factors. These factors are also affected by the local rural areas. Therefore, the construction of rural STEM education system should be consistent but different and maintain its characteristics. It is precisely because unplugged activities can be learned anytime and anywhere without any restrictions. For rural areas with backward economic conditions and unable to be equipped with hardware equipment, it has undoubtedly become an economical choice that can not only learn computer knowledge and promote the development of computing thinking. SWOT analysis method emphasizes the overall analysis, which advocates not only the overall picture and consistency of the unplugged activities used for the cultivation of computational thinking in rural areas from a macro perspective, but also the regional differences and educational reality in rural areas.

3.1. Strengths of unplugged activities in rural areas

To develop unplugged activities in rural areas to cultivate students' computing thinking starts from solving the practical problems faced by rural society and students. It is not limited to programming education and robot education, which are carried out in cities, but to localize the content of STEM education and solve the constraints of the hardware environment for carrying out STEM courses in rural areas. The "unplugged" approach of learning computer principles and knowledge without having to turn on the computer helps the learner stay away from the computer, thinking that children usually regard the computer as a tool or toy, rather than an object of study. "Unplugged" allows learners to avoid the difficulties of going directly to the computer, but to develop computational thinking through tasks or stories related to real situations. That is to say, choosing unplugged activities in rural areas to cultivate students' computational thinking not only achieves the localization of content adapted to rural reality, but also is a way to consider from the characteristics of rural K12 learners, and at the same time restates the fundamental educational purpose of STEM education.

3.2. Weaknesses of unplugged activities in rural areas

Unplugged computational thinking is an activity that tries to solve problems by learning computational thinking in an unplugged way under certain problem situations. We shouldn't think of it as an option to be unable to use a computer, but rather as a powerful complement to the course content. As an activity course, it must form a systematic course content structure if it wants to be promoted in rural areas. However, textbooks on unplugged activities are still lacking, with existing domestic textbooks such as "Unplugged Computer Science" translated from a book of the same name co-published by Tim Bell, Ian H. Witten, and Mike Fellows. We need textbooks that are localized and close to the life of students in stage K12 in China. The classroom practice of unplugged activities is often just a teaching practice attempted by individual teachers, and has not formed a long-term and systematic curriculum system.

3.3. Opportunities of unplugged activities in rural areas

What STEM education embodies is not a single subject, but the internal connection between science, technology, engineering and mathematics. STEM education can enable children to acquire a systematic way of thinking and try to explore more creative learning methods. In rural areas, we have a lot of real problems we can design with. There are too many projects and scenarios in rural areas that can be used to explore the implementation path of STEM courses, such as observing and learning rural water conservancy and irrigation engineering, housing construction engineering, etc. From these projects, small projects in line with the cognitive characteristics of learners are designed to make students realize that a project can not be done by a single person, but by the strength of a team. This opportunity is that we should pay attention to rural construction and agricultural production, take measures according to local conditions, and implement localized unplugged activities such as house construction and crop planting for the construction of new countryside, which can not only directly serve the construction of new countryside, but also help strengthen students' skills of transforming interdisciplinary knowledge into comprehensive practice.

3.4. Threats of unplugged activities in rural areas

STEM education is a new concept, which has a strong flavor of The Times. However, the courses to cultivate computational thinking in rural areas started late in China, especially the STEM courses in rural areas are still in the exploratory stage. Therefore, at the beginning of the implementation of unplugged activities in rural areas, the phenomenon of "empty" is very likely to occur. More activities are "activities" for the sake of "activities", which become the observation of rural construction projects or teacher-centered teaching, but the cultivation of computational thinking is not really implemented. The unplugged activity is just an attempt, and there is no need to do it for the sake of activity. However, the situation and difficulty of the project should be in line with the cognition and development rules of rural learners. Unplugged activities must also be learner-centered. The backward educational concepts and inadequate understanding of STEM education in rural schools may make unplugged activities equal to common comprehensive practice activities. In the face of this threat, we should make it clear that fostering computational thinking should not be an aristocratic curriculum. Rural areas and economically disadvantaged schools need to build curriculum confidence.

4. STRATEGY CONSTRUCTION BASED ON SWOT MATRIX

Unplugged activities in rural areas are affected by a variety of interwoven factors, a factor is not only an advantage, but also may become a disadvantage due to its imperfection. Therefore, it is necessary to straighten out the connection and mutual influence among different influencing factors. In the context of the integrated development of urban and rural compulsory education and education equity, there are both opportunities and challenges for the localized implementation of rural STEM education. According to the SWOT matrix analysis model (Figure 1), there are four strategies to carry out unplugged activities in rural areas as the main form of STEM courses.

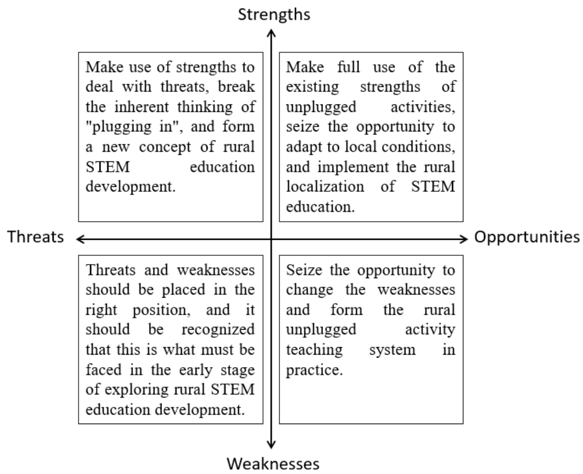


Figure 1. SWOT Matrix Analysis of Unplugged Activities in Rural Schools.

Advantages and opportunities are the positive influencing factors in the process of unplugged activities in rural areas. The advantages mainly come from the unplugged activities themselves, while the opportunities mainly come from the local characteristics of rural areas. Threat and disadvantage are negative influencing factors. The threat mainly comes from the incomplete research and practice of unplugged activities, and the threat mainly comes from the backward concept of rural STEM education. The SO strategy emphasizes the use of advantages and opportunities, but pays insufficient attention to disadvantages and threats. The SO strategy is to implement path localization. Implement STEM curriculum from rural environment is not only scientific and feasible, but also helpful for students to experience the great significance of technology in agricultural production, and guide students to understand the comprehensiveness, practicality and innovation of STEM curriculum. Researchers will explore more types of unplugged activities and conduct multiple rounds of practice on a larger scale to ensure the efficient advancement of STEM courses. Although the SO strategy is the most ideal strategy, it is impossible to have only advantages and opportunities in reality. The grasp of disadvantages and threats is also the key to strategic decision-making and planning. The ST strategy focuses on the relationship between advantages and threats. The ST strategy is to break the inherent thinking, the formation of a new concept. "Unplugging" is the means, while promoting the development of learners' computational thinking is the goal of education. Don't assume that "unplugged activity" doesn't satisfy your cognitive needs. Instead, unplugged activity is the most appropriate choice based on the situation. It does not increase cognitive load and lays the foundation for complex computer science concepts and techniques. The development of rural STEM education may still have a long period of preliminary exploration, and we need to start from the formation of a new STEM teaching concept. The WO strategy focuses on hedging against adverse factors. The WO strategy is to form a rural unplugged activity teaching system in practice. It is a very worthy direction to develop students' computing thinking through unplugged activities, which requires researchers to design content systems in line with learners' development characteristics.

5. SUMMARY AND SCOPE

Currently, computer programming courses in primary and secondary schools in China's urban areas are in full swing, but are rarely covered in rural primary and secondary schools due to the lack of hardware equipment. In the new era, computing thinking should become an essential skill for every learner, which has realized students' good yearning for quality education. Computational thinking is no longer only a method to solve problems by using information technology tools, but also a thinking process to analyze, understand and deal with real life problems. Rural areas should localize STEM education according to local conditions. As an economical and affordable way to learn computer knowledge and promote the development of computational thinking, unplugged activities will help rural primary and middle school students cultivate their computational thinking. Through the methods of expert guidance, follow-up guidance and practice exploration, the best methods and modes of unplugged activities in rural primary and secondary schools are gradually explored to create a "template" and explore innovative "ways". Through the joint efforts of all of us, we will actively explore the STEM education methods and strategies suitable for rural development, and serve more rural frontline teachers and children.

6. REFERENCES

Bell, T., Witten, I.H., & Fellows, M. (1999). *Computer Science Unplugged: Off-Line Activities and Games for All Ages*: http://csunplugged.org.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2019). *Computer science unplugged: School students doing real computing without computers*: *The New Zealand Journal of Applied Computing and Information Technology* (1), 20-29.

Dou Y. (2015). *Application Research of Unplugged Computer Science Teaching in Information Technology Classroom of Primary and Secondary Schools: Information Technology Education in China*(Z1), 187-188.

Li J. & Yao J. (2020). *A study on the equality of opportunities in STEM education : Science of Contemporary Education*(08), 16-22.

Ministry of Education. (2017). *Notice of the Ministry of Education on the issuance of the Curriculum Plan and Subject Curriculum Standards for Regular Senior High Schools* (2017 Edition) http://www.moe.gov.cn/srcsite/A26/s8001/201801/t20180115_324647.html

Wei Y. (2015). *Unplugged computer science II , haveyou tried it?: Information Technology Education in China* (11), 18.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Xi J. (2017). *Decisive Victory in Building a Moderately Prosperous Society in All Respects and Achieving the Great Victory of Socialism with ChineseCharacteristics for a New Era -- Report Delivering at the 19th National Congress of the Communist Party ofChina*: People's Publishing House.

Zhang S. (2021). *Development and Implementation ofLow-cost STEM Curriculum in Rural Primary Schools* :Shi Dao(01), 53-54.

Zeng M. (2020). *Strategies and Path of Localization of STEAM Curriculum in Rural Junior Middle Schools : Teaching and Management*(09), 92-94.

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Computational Thinking Implementation in Schools – An Experience with Rural Welfare Schools in India

Pooja PALAPARTHI
Andhra Pradesh Social Welfare Residential School for Girls, Narsipatnam, Visakhapatnam District, India
poojapusphasri@gmail.com

ABSTRACT

The Computational Thinking curriculum has been introduced since Academic year 2018-19 across Classes 5-12 in 427 schools with 200,000 students across 13 Districts in Andhra Pradesh, a state in India. This paper describes the approach used in implementation of unplugged activities to help students develop a systematic approach to problem solving through localization to overcome the challenges of language and culture.

KEYWORDS

Unplugged, Computational Thinking in K12

1. INTRODUCTION AND BACKGROUND

The Computational Thinking(CT) curriculum is being taught to 200,000 students in 427 APSWREIS (Andhra Pradesh Social Welfare Residential Educational Institutions Society) Social Welfare, Tribal Welfare and Ashram schools across classes 5 to 12 in partnership with CSpathshala since 2018. The goal of this partnership is developing Computational thinking, explorative skills and reasoning abilities in rural students who are traditionally deprived through a customised CT curriculum and providing pedagogical as well as content training to teachers.

Of these the Social Welfare Residential Schools were started in 1983 by the Government of Andhra Pradesh (AP) with the objective of providing quality education to the children belonging to the Scheduled Caste, Scheduled Tribes and other backward classes of the Society. There are 189 Social Welfare schools across all 13 districts of AP catering to 106,783 students from class 5th to 12th, majority are first-generation learners belonging to marginalized communities with an annual family income of less than Rs. 100,000 (1400 US$). Of these, 123 are Girls schools and 66 are Boys schools.

CSpathshala has been working in partnership with APSWREIS to bring computational thinking to these schools and prepare students for the digital age .

CSpathshala (www.cspathshala.org) is an Association for Computing Machinery (ACM) India education initiative to bring a modern computing curriculum to Indian schools. CSpathshala has developed a class room, activity based Computational Thinking Curriculum for K-12. CSpathshala has adopted the unplugged activity-based approach to teach Computational Thinking (CT) without the use of computers.

Initially, I worked in Tribal Welfare Residential school (Girls), Araku valley located at Visakhapatnam district, Andhra Pradesh state where students are from tribal areas. The school is situated in Araku Valley is a tribal area in the interior 120kms away from district Head Quarters with no internet connectivity and limited public transport makes access difficult. The primary school gives them limited exposure with basic literacy skills. These tribal girls in grade 5 would speak only the local dialect and could not understand Telugu. Since the native tribal languages are varied, I took help from other students and devised the communication strategy with them.

Post the Tribal school assignment, I got transferred to Social Welfare residential school (Girls) at a location named Narsipatnam which is located in the rural area of Visakhapatnam district where students from surrounding villages attend their schooling here. Total students strength in the school is around 621 and belongs to the grades of 5th till 12th. Total allotted teaching hours per week for Computational Thinking are 17 in the school and I am the only assigned trainer handling Computational thinking concept in the school.

CSpathshala team had conducted a Baseline Assessment Test for 663 students from 11 APSWR schools from 9 Districts, April 2018 to compare the comprehension and analytical skills of the students to appropriately customize CSpathshala computational thinking curriculum for APSWR. The assessment test was designed to test the Computer Science (CS) domain for Computational Thinking (CT) skills in Decomposition, Patterns, Abstraction and Generalization, Algorithms and Evaluation. Based on the findings, CSpathshala developed a customised Curriculum and Implementation Plan for APSWR Standards 5-12 which was shared in Jul y 2018. A bridge course was designed for Standards 5-12 for First Year (2018-19) of Implementation with a subset of lessons to provide a foundation for CT in students before the commencement of the full syllabus. From the second year APSWR planned to have 1 period per week for CT - a total of 32 periods and centrally the syllabus and the monthly lesson plans were shared with all the 427 schools.

2. IMPLEMENTATION OF CT IN SCHOOLS

The first training was conducted in July 2018 by the CSpathshala team with various activities like Sudoku, patterns, counting combinations and I was enthusiastic about trying out these activities in the classroom with our students. We got guidance from 2 mentors who emphasized us to find real time examples that are easily relate to the computational thinking concepts. I also started realizing that the students grasp the concepts and understand better only when there are good and relevant real life examples. With the motivation from my fellow Trainers of other schools and Master Trainers, I learnt some good practices with examples for implementing CT successfully at classroom level.

Since I am working as IT & CT Trainer in these schools, for the past 3 years now it has given a good awareness of the students levels in areas such as their memory, grasping power through the teaching techniques. While I am teaching theory classes like introduction to computers, history of computers, generations of computers students find it monotonous and got bored of those classes. This made me explore various rhymes and games online related to computers which they eventually started enjoying. Training students on concepts which are not based on any hardware systems is not only challenging but different. Hence, I decided to impart the CT concepts by making them learn through performing various activities and ignite their logical thinking.

The 2019-20 academic year, I got at least one innovative example per computational thinking concept which also made students stay motivated and gave them a platform to be more participative in the class. Students come from economically backward families and the majority of the parents are engaged in agricultural activity. Below are some accounts of the approach used in my classroom:

3. SUDOKU

It is a familiar game with students that boosts logical thinking, improves memory and be analytical but most of them were not following systematic approach to solve Sudoku. This made me explain the approach to solving Sudoku with some activities that used examples such as different types of fruits.

In the CT lab, Sudoku was presented using the Sudoku board which was prepared by the students in the previous academic year. On No Bag Day we conducted a competition between different groups to solve 4 * 4 Sudoku by using a systematic approach. The team who solved Sudoku in less time will score the point.

4. ALGORITHMS: GUESS MY BIRTHDATE ACTIVITY

This activity is conducted with the teacher who is a “robot” and can answer the questions with a “Yes” or a “No” and the students are required to guess the birthdate. This activity helps students develop an algorithm to guess the birthdate using binary search.

In the introduction class I started with basic topics like even number, odd number, division and then I prepared some circles and marked them with 1 to 31. By using those circles I let them do activity by asking questions. They interacted well and started guessing their friends’ birthdays. Some students played with their friends in class while participating in that they enjoyed a lot.

The students of 8th class didn't understand the concept of guess my birthdate, what is a good question and they were confused. Then I prepared some papers by marking with numbers 1 to 31 and asked them to come down to the playground and asked 31 confused students to form a line. Each student was given a paper and the other students asked them the questions to guess the birthdate.

After that they came to know what the concept is and how to guess birthdate using binary search method. Then after students started playing with circles to guess their friends' birthday. The approach to this is to constantly be cutting the amount of numbers you have to consider in half. This game is a simple example of a binary search process!

5. CT ANALYSIS AND IMPACT

The students shared their feedback that they enjoyed the activity of 4x4 Sudoku using the boards in the CT lab more than solving Sudoku using pen and paper. After that. we also conducted a Sudoku competition in the assembly area for all classes 5th to 10th. None of the students faced any difficulties in solving the Sudoku puzzles. All the students learnt to solve Sudoku using a systematic approach to problem solving and moved from 4x4 Sudoku to 6x6 and 9x9 Sudoku with ease.

Sudoku challenge was conducted across 427 schools for classes 5 to 9 with a participation of 120,000 students, perhaps the largest Sudoku challenge ever.

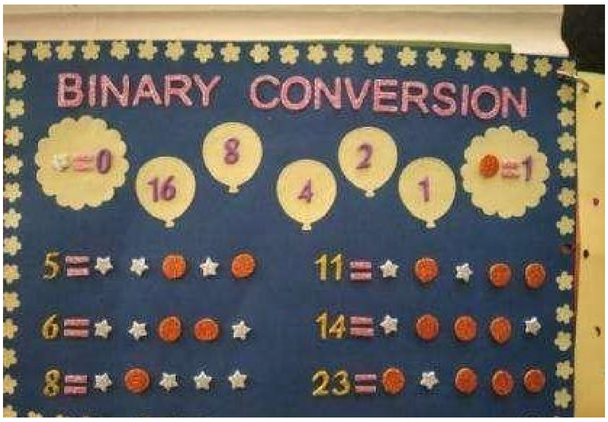
I collected feedback from the students through interaction and my findings are based on classroom observation. While conducting guess my birthdate for 8th standard I noticed that 50% of the students did not understand the activity, I chose the approach of conducting a physical activity with the confused students to demonstrate "what is a good question" and elimination so that they can guess the birthdate. After conducting this activity all the students understood the systematic approach and they started conducting this activity with their friends too!

Students shared that they enjoyed the CT activities. I encouraged the girls to share additional examples for the CT concepts, counting combinations and patterns which we used in the classroom that motivated students in sharing and learning concepts through innovative ideas.

We showcased the activities and Scratch projects to parents during their regular monthly visits to school. We also created an activity book with the students for all the interesting activities conducted.

6. ACTIVITY BOOK

The use of art and craft gave the students an opportunity to apply their newly learned CT skills by creating these activity books. The activity book is like a repository that includes topics like Counting Combination, Sudoku, Missing Number Pyramid, Sorting, Patterns in Words, Pattern in Numbers, Pattern in Shapes, Morse Code, Odd Man Out and Binary Conversion). We made it with simple materials that are easily available such as foam sheets, colored papers and some stones which are easily available. This activity book will act as a ready reckoner and explain the gist of each concept powerfully.



4. CONCLUSION

In an overall feedback given by students and parents, they are feeling very happy for the implementation of CT through this project in schools. During the monthly visits written feedback was collected from the parents. They shared that the students will find CT useful in appearing for competitive exams as CT has helped students develop problem solving skills. The localization of examples invoked an interest amongst students and they enjoy learning CT through these unplugged activities. This was also visible through the increased participation of students in using art and craft to demonstrate CT activities. Using CT activities and relating to real life examples is helping them experience the concepts better in their journey to making them Future ready.

5. BIBLIOGRAPHY/REFERENCES

Computational Thinking in K-12 Education: https://cspathshala.org/2017/10/25/computational-thinking-curriculum/

Computer Science Unplugged: csunplugged.org/

CSpathshala syllabus and CSpathshala curriculum documents: https://cspathshala.org/curriculum/

Teaching aids Resources from www.cspathshala.org

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Computational Thinking and Unplugged Activities: Localization Enabling Learning

Lakshmi Durga PETTA

Andhra Pradesh Social Welfare Residential School for Girls, Godi Village, East Godavari District, India.

durgalakshmi765@gmail.com

ABSTRACT

This paper demonstrates the use of a local pyramid game to teach students the concepts of systematic counting, listing and reasoning through the missing numbers activity. A specific One of the classroom activities described in this paper is Number challenge and Pyramid puzzle with a goal of students learning how to solve 2 types of puzzles using a systematic approach to problem solving. As this activity requires only single digit addition and subtraction, it is an activity all students can participate in 5th grade .

KEYWORDS

Unplugged, Computational Thinking in K12

1. INTRODUCTION AND BACKGROUND

The Computational Thinking (CT) curriculum is being taught to 200,000 students in 427 APSWREIS (Andhra Pradesh Social Welfare Residential Educational Institutions Society) Social Welfare, Tribal Welfare and Ashram schools across classes 5 to 12 in partnership with CSpathshala since 2018. The goal of this partnership is to develop Computational thinking, explorative skills and reasoning abilities in rural students who are traditionally deprived through a customized CT curriculum and providing pedagogical as well as content training to teachers.

Of these the Social Welfare Residential Schools were started in 1983 by the Government of Andhra Pradesh (AP) with the objective of providing quality education to the children belonging to the Scheduled Caste, Scheduled Tribes and other backward classes of the Society. There are 189 Social Welfare schools across all 13 districts of AP catering to 1,06,783 students from class 5th to 12th, majority are first-generation learners belonging to marginalized communities with an annual family income of less than Rs. 100,000 (1400 US$). Of these, 123 are Girls schools and 66 are Boys schools.

CSpathshala (www.cspathshala.org) is an Association for Computing Machinery (ACM) India education initiative to bring a modern computing curriculum to Indian schools. CSpathshala has developed a classroom; activity based Computational Thinking Curriculum for K-12. CSpathshala has adopted the unplugged activity-based approach to teach Computational Thinking (CT) without the use of computers.

CSpathshala team had conducted a Baseline Assessment Test for 663 students from 11 APSWR schools from 9

Districts, April 2018 to compare the comprehension and analytical skills of the students to appropriately customize CSpathshala computational thinking curriculum for APSWR. The assessment test was designed to test the Computer Science (CS) domain for Computational Thinking (CT) skills in Decomposition, Patterns, Abstraction and Generalization, Algorithms and Evaluation. Based on the findings, CSpathshala developed a customized Curriculum and Implementation Plan for APSWR Standards 5-12 which was shared in July 2018. A bridge course was designed for Standards 5-12 for First Year (2018-19) of Implementation with a subset of lessons to provide a foundation for CT in students before the commencement of the full syllabus. From the second year APSWR planned to have 1 period per week for CT - a total of 32 periods and centrally the syllabus and the monthly lesson plans were shared with all the 427 schools.

Our school was established on 22.11.2005 for economically backward students. These students are first generation learners from remote villages areas and they attend local government primary schools. Majority of the parents are farmers, daily wagers are not educated. We have 553 girl students from 5th to 12th Grade attending this residential school. I am the only resource person who has been appointed as an IT & CT (Information Technology and

Computational thinking) trainer in our school for implementing CT across these grades.

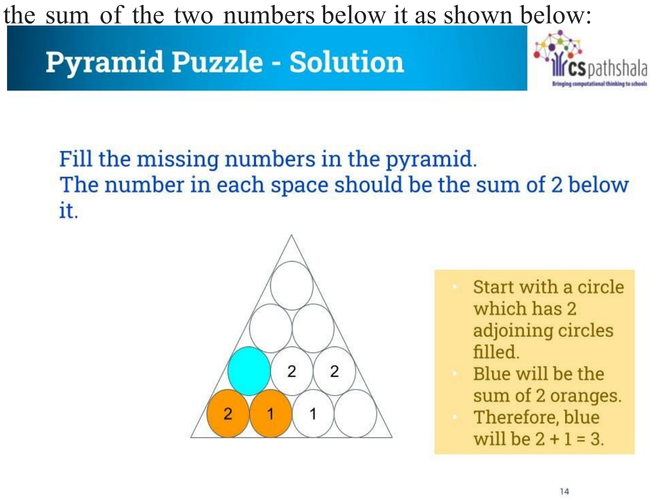
2. IMPLEMENTATION OF CT ACTIVITIES IN SCHOOLS

I attended my first training program in July 2018 on Computational Thinking which was conducted by the CSpathshala team with various activities like Sudoku, patterns, counting combinations. During the training we used various objects like bottles and glasses to demonstrate how these could be related to daily life activities and would make learning fun for students through these unplugged activities.

I was enthusiastic about trying out these activities in the classroom with our students for teaching CT.

3. PYRAMID PUZZLE

Pyramid puzzle is an activity that helps children to learn problem solving skills. The students learn how to use basic addition and subtraction to solve the number pyramid. As per the learning objectives, each student has to solve a pyramid puzzle by filling the missing numbers, so that the numbers in each circle should be the sum of the two numbers below it as shown below:



First, I facilitated learning this concept with the students of the 5th Grade (10-11years old) in their notebooks. The children felt that these are the missing numbers and filled it in their notebooks, and felt that the same task was being conducted repetitively. As a next step, to add more clarity to solving this puzzle to the students, I decided to try this differently as a game.

I thought of this idea for implementing the pyramid puzzle using drinking water glasses from my visit to the Village festival and exhibitions and not any book or internet as a resource. I remember the pyramid game that I would play where drinking water glasses are arranged as a pyramid and you have to strike all of them down with game balls in three chances. I used this idea for creating a game for the number pyramid.

For this pyramid puzzle as a game I used some commonly available materials like paper glasses and sticky notes. I wrote the given numbers on the sticky notes and pasted them on the paper glasses. I also drew the pyramid puzzle with chalk on the floor in the classroom. I asked the students to divide themselves into two teams. Then, I kept the paper glasses with the numbered sticky notes in front of them as shown below:

Next I asked them to make the pyramid using the paper glasses by arranging them using the rules of the number pyramid ie. The number on the top glass has to be the sum of the two glasses immediately below it... Here they had to identify the particular numbered glass and form a pyramid which will also help them to fill the pyramid that was drawn on the floor.

The team that made the pyramid with the paper glasses first got the chance to fill the pyramid which was drawn on the floor with a piece of chalk. This team is the winner, and the other team is the runners - up.

4. ANALYSIS AND IMPACT

I collected the feedback through interactions with the students and in classroom observation. Initially the students were hesitant and shy to participate in the activity. I got some of the enthusiastic girls to try the game and seeing them enjoy it, all the students wanted to participate and asked me to conduct the activity again.

When I implemented this activity, through the physical game using the paper glasses, the students enjoyed and also showed enthusiasm to take part and complete this game. They learned solving the puzzle with fun and I felt immensely happy when I saw their increased participation and interest levels.

This activity also helped in improving the students' ability in doing addition and subtraction. I have noticed that students improved the stepwise approach while solving this activity multiple times. Students didn't use trial and error methods to solve the pyramid but unknowingly developed the strategy to win the game by using a systematic approach. My observation was that the time taken by the students to solve the number pyramid was less with the game in comparison to solving the number pyramid in their notebooks.

I would like to share that this activity helped to improve the reasoning ability of students and developed counting skills. Students exhibited their enthusiasm to participate, use visualization and pattern formation using Pyramids. The students learnt problem solving with fun.

5. CONCLUSION

The feedback from the students was very encouraging as it helped them develop skills in systematic listing, counting and reasoning and overcome the fear of numbers. The use of the game invoked an interest amongst the students and they enjoy learning CT through these unplugged activities. This was also visible through the increased participation of students in participating in the game with a positive spirit of competition CT activities.

6. BIBLIOGRAPHY/REFERENCES

Computational Thinking in K-12 Education: https://cspathshala.org/2017/10/25/computational-thinking-curriculum/

Computer Science Unplugged: csunplugged.org/

CSpathshala syllabus and CSpathshala curriculum documents: https://cspathshala.org/curriculum/

Teaching aids Resources from www.cspathshala.org

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

運算思維教育桌遊與圖形化程式設計對初學者學習運算思維之影響

楊士弘 [1]，許庭嘉 [2*]，陳沐生 [3]
[1]臺北市立大同高級中學，臺灣
[2, 3] 國立臺灣師範大學科技應用與人力資源發展學系，臺灣

yangtwosu@hotmail.com，ckhsu@ntnu.edu.tw，mushengchen946@gmail.com

摘要

本研究旨在透過教學實證研究探討初學者的七年級學 生使用運算思維教育桌遊的實驗組共 85 人，和使用圖形化程式設計 Scratch 的控制組共 84 人，二組使用其不同媒體完成六節課結構化程式設計的邏輯練習之 後，其運算思維能力和外在認知負荷表現為何。結果 以運算思維概念之前測作為共變項、後測作為依變 項，經共變數分析結果實驗組學習成效顯著高於控制 組。另外，在六週實驗之後的實驗組與控制組外在認 知負荷獨立樣本 t 檢定結果顯示實驗組的外在認知負荷低於控制組，可見運算思維教育桌遊比圖形化程式 設計更適合初學者入門使用。

關鍵字：

運算思維；教育桌遊；圖形化程式設計；外在認知負 荷；結構化程式設計

1. 前言

近年來圖形化程式設計，又稱「積木程式設計」工具，例如：Scratch、App Inventor... 等一直是教師用來帶領學生入門的工具。隨著台灣 108 課綱的推出，國中程式設計的教學核心是運算思維的訓練，不再是 只有學習程式碼本身，更進一步的要以問題解決主軸，培養學生利用資訊科技與運算思維解決問題之能 力（教育部，2019）。

隨著這波全民運算思維教育的熱潮，為了因應教師們 教授新課程的需求，各式各樣的教學工具也不斷地被 開發出來，「新機器人蓋城市-Robot City v2」便是其中一種運算思維教育桌遊的產品（楊士弘，2020）， 該產品也將被納入七年級正式資訊科技教育部審定本 教科書的一部分。這是一套以運算思維與結構化程式 設計邏輯為基礎所開發的教育桌遊，希望能以「不插電」的形式和遊戲式學習的規劃，讓學生在不使用資 訊設備時也可以學習運算思維。

過去曾有研究試著將此運算思維教育桌遊融入課程 中，發現學生對於使用桌遊來學習程式設計概念的模 式反應良好（楊士弘、許庭嘉，2020），而本研究將 進一步的探討使用 Robot City v2 和 Scratch 對運算思維能力初期養成的成效比較。

2. 文獻探討

2.1. 運算思維桌上遊戲

在近幾年，運算思維與其相關的概念，例如:程式編寫、程式設計、演算法等等，在教育領域中越來越受到關注(Bocconi et al., 2016)，這是新一代學生他們必須具備的能力，不管在哪一種領域，他們都必須擁有運算思維的問題解決能力，才能在現代環境中有所發展(Román-González et al., 2017)，因此，運算思維儼然成為每個人必備的基本技能(Yadav et al., 2014)，如何提高學生運算思維的能力變得至關重要。近年來已有許多研究人員將桌遊融入教學中，Gee (2005)表示透過確立學習目標，桌遊可以成為促進自我導向學習、問題解決及深度學習的有效工具之一，學生可以透過社交、與同儕互動及相互學習來遊玩教育桌遊(Wu et al., 2014)，以此顯示教育桌遊具有支持互動學習的獨特優勢(Mayer & Harris, 2010)。Kuo et al.(2020)將運算思維教育融入教育桌遊中，透過桌遊所具備的競爭性與合作性，成功地提升學生學習運算思維的成效。

2.2. 運圖形化程式設計

近年來，已有越來越多的研究將電腦科學整合到國中 小的教育之中，學生已經提早開始學習運算思維相關 的知識，但是由於文字型的程式教學，許多學生認為 是非常難以學習的(Wilson & Moffat, 2010)，所以必須

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021*. Singapore: National Institute of Education.

要找出其他的方式來進行程式教學。因此使用圖形化程式可以解決許多的問題，並且可以讓年齡更小的學生進行學習。學生透過移動程式積木來撰寫程式，積木堆疊擺放的過程，提供學生對於程式撰寫的直覺性(Maloney et al., 2010)，可以避免常常從文字型程式獲取錯誤訊息的失落感，這是對於入門程式教學的極大助益，並且可以透過圖形化程式教學，來擺脫以往傳統文字型程式教育艱澀難懂的過程，這對於學生的學習有效性至關重要(Wilson & Moffat, 2010)。

3. 研究方法

3.1. 研究對象

本研究的研究對象為臺北市某國中的七年級學生，6 個班級共 169 人，並隨機取 3 個班級共 85 人做為實驗組，3 個班級共 84 人做為控制組。透過前測掌握受測者於教學實驗之前的起始能力，然後實驗組使用運算思維教育桌遊做為學習媒介，控制組使用 Scratch 做為學習媒介進行教學實驗。

3.2. 評估工具

本研究中前測和後測的考卷是由二位教師取用結構化程式設計和運算思維評量之題目並修訂完成，共 9 題與流程圖相關的選擇題，以及 11 題取自 Bebras 國際運算思維挑戰賽的測驗題，每題 5 分，前、後測題目相同但是呈現順序不同。

認知負荷量表採用的是 Hsu (2017)的同一份測量外在認知負荷 - 心智努力(Mental Effort)的量表題目依據認知負荷理論(Sweller, Van Merrienboer, & Paas, 1998)總共有 3 題，其信度 Cronbach α 值 0.85。

3.3. 實驗設計

本次實驗共進行 6 節課，每節課 50 分鐘，共 300 分鐘，流程與內容簡介於表 1。

表 1 、實驗流程簡介

節	實驗組(桌遊)	控制組(程式)
1	前測、演算法簡介	
2	流程圖教學	
3	循序、選擇結構教學	循序結構教學
4	重複結構教學	選擇結構教學
5	綜合練習	重複結構教學
6	後測	

第一堂課先進行前測，最後 15 分鐘的時間向學生簡介演算法和結構化程式設計的概念，第二堂課搭配課本的內容做流程圖的教學與練習。第三至五節課兩組有不同的操作，實驗組會使用運算思維教育桌遊對戰來練習結構化程式設計的邏輯和經歷運算思維的演練，如圖 1 所示。控制組則搭配傳統課本內容，以 Scratch 專案的方式帶領學生完成練習結構化程式設計的邏輯和經歷運算思維的演練，如圖 2 所示，最後第六堂課則再對兩組的學生進行後測。

圖 1 實驗組使用運算思維教育桌遊進行結構化程式設計教學活動

圖 2 控制組使用 Scratch 專案進行結構化程式設計教學活動

4. 研究結果

4.1. 學習成效分析結果

本研究使用共變數分析來量測實驗組與控制組的學習成效，同質性檢定未達顯著(p>0.05)，表示實驗組與控制組之間的起始能力並無差異，根據表 2 得知，實驗組與控制組之學習成效有顯著差異($F=17.030^{***}$, $p<0.001$)，實驗組的後測成績（$M=44.941$，$SD=11.840$）顯著高於控制組的後測成績（$M=36.905$，$SD=13.261$），由此可見，學生使用桌遊來學習運算思維的成效比使用 Scratch 還要好。

表 2、學習成效共變數分析

組別	人數	平均值	標準差	調整後平均數	標準誤	*F*
實驗組	85	44.941	11.840	44.930	1.370	17.030***
控制組	84	36.905	13.261	36.909	1.378	

****p*<0.001

4.2. 外在認知負荷分析結果

本研究也對於外在認知負荷進行探討，使用獨立樣本 t 檢定分析，根據表 3 得知，實驗組與控制組之外在認知負荷有顯著差異(*t=-3.022*^**^*, p<0.01*)，實驗組的外在認知負荷（*M=2.973*，*SD=1.546*）顯著低於控制 組的外在認知負荷（*M=3.694*，*SD=1.559*），由此結 果顯示，學生使用桌遊來學習運算思維的外在認知負 荷比使用 Scratch 學習運算思維的外在認知負荷更低。

表 3、外在認知負荷之獨立樣本 t 檢定分析

組別	人數	平均值	標準差	t
實驗組	85	2.973	1.546	*-3.022***
控制組	84	3.694	1.559	

***p*<0.01

5. 討論與結論

5.1. 討論

未成年的學生都是喜愛玩遊戲的，或許「新機器人蓋 城市-Robot City v2」桌遊原先設計的方向就是以培養運算思維和程式設計的概念出發(Kuo, & Hsu, 2020)，加上上課能以和同學一起玩遊戲兼學習的方式進行， 不只能強化結構化程式設計的概念而且有經歷運算思 維歷程的演練機會，故讓桌遊在本次研究的學習成效 與認知負荷中，在初學者早期有較好的成效表現。

5.2. 教師反思

根據本次的研究，我們可以得知使用運算思維桌遊進行教學，不僅在運算思維初學者的學習成效上優於更早提出的圖形化程式設計教學，學生的認知負荷也較圖形化程式設計教學來得低。在學生以桌遊入門達到好的成效後，期盼後續研究能探討組合桌遊和 Scratch 相關的教學，讓學生能有效地學習更深入的程式設計概念。

5.3. 研究限制與未來研究

在小學的資訊課程中，大多數的學生皆有學習過程式 設計，因此我們並不能夠找到對程式設計完全沒有先 備經驗的學生來進行實驗，僅能使用共變數分析來排 除兩組間的差異，此為本研究的限制。未來研究的方 向建議可討論混合 Scratch 和 Robot City v2 的教學模式，或延長實驗時間，以探討如何結合插電與不插電 的教學，將能有更好的學習成效。

6. 致謝

本研究感謝科技部研究計畫編號: 108-2511-H-003 -056 -MY3 補助。

7. 參考文獻

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., &Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (No. JRC104188). Joint ResearchCentre (Seville site).

Gee, J. P. (2005). Learning by design: Good video games as learning machines. *E-learning and Digital Media, 2*(1), 5-16.

Hsu, T.-C. (2017). Learning English with Augmented Reality: Do learning styles matter? *Computers & Education, 106,* 137-149.

Kuo, W.-C.,& Hsu, T.-C. (2020). Learning computational thinking without a computer: How computational participation happens in a computational thinking board game. *The Asia- Pacific Education Researcher, 29,* 67-83.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education, 10*(4), 1-15.

Mayer, B., & Harris, C. (2010). *Libraries got game: Aligned learning through modern board games*. Chicago, Illinois: American Library Association.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678-691.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational psychology review*, *10*(3),251-296.

Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to Introduce Younger Schoolchildren to Programming. *Psychology of Programming Interest Group, 1*(1), 1-12. Retrieved February

20, 2021, from https://www.ppig.org/files/2010-PPIG-22nd-Wilson.pdf

Wu, C.-J., Chen, G.-D., & Huang, C.-W. (2014). Using digital board games for genuine communication in EFL classrooms. *Educational Technology Research and Development, 62*(2), 209-226.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1), 1-16.

教育部（2019）。***十二年國民基本教育課程綱要科技領域***。2021 年 3 月 15日取自 https://cirn.moe.edu.tw/Upload/file/27526/6641 2.pdf

楊士弘、許庭嘉（2020）。***使用運算思維教育桌遊實踐不插電教學—以「新機器人蓋城市」 桌遊為例。*** 2020 年第四屆計算思維教育國際會議發表之論文，香港會議展覽中心。

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

The Effects of Computational Thinking Educational Boardgame and Visual Programming on the Novices Learning Computational Thinking

Shih-Hung YANG[1], Ting-Chia HSU[2*], Mu-Sheng CHEN[3]
[1] Taipei Municipal Datong High School, Taiwan
[2, 3] National Taiwan Normal University, Taiwan
yangtwosu@hotmail.com, ckhsu@ntnu.edu.tw, mushengchen946@gmail.com

ABSTRACT

The purpose of this study is to explore the computational thinking competence and external cognitive loads of the novices who are 7th grader students through the empirical instructional research. The instructional experiments were involved 85 students in the experimental group using the educational board game of computational thinking, and involved 84 students in the control group using the visual programming tool which is Scratch. Both groups experienced the same logical training of structure programming with different the respective media, which took 6 periods in total. The covariance was the pre-test of computational thinking, while the dependent variable was the post-test of computational thinking. After the analysis of ANVOCA, the results showed that the experimental group outperformed the control group. In addition, after the six-week instructional experiments, the external cognitive loads of the experimental group were significantly lower than that of the control group based on the results of the independent t-test between the two groups. Accordingly, the computational thinking educational boardgame is relatively suitable for the novices in comparison with the visual programming tool.

KEYWORDS

Computational thinking, educational board game, visual programming, external cognitive load, structural programming

Computational Thinking and Subject Learning and Teaching in K-12

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Designing a Computational Thinking Curriculum for Everyone with a Differentiated and Gamified Approach

Phylliscia CHEW[1*], Da LI[2*]
[1,2]NUS High School of Mathematics & Science, Singapore
nhscsp@nus.edu.sg, nhsld@nus.edu.sg

ABSTRACT

Computational Thinking "involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing, 2006: p. 33). In NUS High school, we believe Computational Thinking is a fundamental skill for everyone, not just for interested students. Every 21st century student should have the opportunity to learn about Computer Science. Hence, every student in our school will have to take a compulsory Computational Thinking module. Via this semester-long module, students will be exposed to three key areas in Computer Science: Unit 1) Problem Solving and Algorithms, Unit 2) Programming Principles & Concepts and Unit 3) Basic Data Skills. In this paper, we will share our experience in designing the Computational Thinking curriculum for everyone, with a differentiated and gamified approach, to cater to students of various learning abilities and interests.

KEYWORDS

Computational Thinking, Curriculum Design, Differentiation, Gamification, Problem Solving

1. INTRODUCTION

We believe that Computer Science will help train students' Logical Thinking, Problem Solving skills, Creativity and Critical Thinking. The six-year Computer Science curriculum in NUS High focuses on the study of Problem Solving & Algorithms, Programming Concepts & Principles, Data Skills and Application Development, and is divided into two key stages – Foundation Years and Specialisation Years. In the Foundation Years (Year 1 to 3), students are exposed to a breadth of topics in Computer Science so that they can appreciate what the study of Computer Science is about. In particular, all students will be required to read a compulsory Computational Thinking module in Year 1 Semester 2. The module aims to ignite students' interest and passion in Computer Science, and also serve as a foundation for modules offered in the later years. In the Specialization Years (Year 4 to 6), students will be exposed to more advanced Computer Science concepts such as Artificial Intelligence and Computer Networking, and relate these ideas to the diverse computing systems and applications in real life.

2. CURRICULUM DESIGN

We adopted and modified the four cornerstones of computational thinking framework presented by BBC bitesize, namely decomposition, pattern recognition, abstraction and algorithm design. Table 1 shows an overview of the Computational Thinking module, and where each CT concept is applicable.

Table 1. Overview of Computational Thinking Module.

Unit	Chapter Outline	CT Concepts
1	Problem Solving Algorithms Data Representation	decomposition, pattern recognition, abstraction, algorithm design
2	Looping with Turtles Computation with Python	decomposition, pattern recognition, abstraction, algorithm design
3	Data Cleaning Data Analysis Data Visualisation Data Security	pattern recognition, abstraction, algorithm design

2.1. Unit 1) Problem Solving and Algorithms

The first unit of the module is on Problem Solving and Algorithms, where the former is one of the most important skills for a computer scientist. The process of problem solving, in the context of computational thinking, starts with understanding and defining the problem, followed by brainstorming possible solutions, iteratively refining and reviewing solutions, and finally, expressing the solution clearly and accurately. We will illustrate with one example:

"The computer is going to randomly select an integer from 1 to 15. You have to guess the number by making guesses until you find the number that the computer chose. How many guesses do you need to always get the correct answer?"

Based on the feedback given to each guess, both cases of linear and binary search are discussed in class. Students will be exposed to the concept of complete search and divide and conquer, where the latter is one approach under decomposition. The computer science terminologies used have been adjusted to suit the cognitive level of the students. For example, instead of using "the worst case", the phrase "*always get the correct answer*" is used. Mathematical calculations, terms and expressions commonly used in time complexity analysis are not used here.

We believe that it is essential to train students to think critically and analyze the problem thoroughly before they embark on programming. Thus, in the design of the Computational Thinking module, we start students off with a series of unplugged problems so that they can learn to:

- identify the important details needed to solve a problem (*in the above example, it is the feedback given to each guess. For the case of binary search, the feedback given is "too large", "correct" or "too small" – abstraction is applied here*),

- break a problem down into small, logical steps (*the range to guess could be narrowed down to either left portion i.e. numbers smaller than the guess or right portion i.e. numbers larger than the guess - pattern recognition and decomposition are applied here*),
- use these steps to create a process (algorithm) that solves the problem, (*First, to guess the middle value in the list. Next, move to the portion where the answer is located. Finally, repeat the process until the answer is found – the process of algorithm design is implicitly carried out here.*),
- and finally, evaluate this process (*we will need 4 guesses to always see the "correct" answer in this example*).

In the above example, "*from 1 to 15*" can be generalized to "*from 1 to N, where N is a positive integer"*, which brings us to the next chapter on algorithms. The formal way to concisely describe the solution of a generalized version of the question in the problem solving chapter is using an algorithm. An algorithm is executed by a computer through programming. Hence, this key chapter on algorithms acts as a transition from problem solving to programming, allowing students to draw the connection between problem solving, algorithms and programming. The introductory examples in the algorithms chapter are extensions of those used in the problem solving chapter. The definition, properties and examples of algorithms are covered, as well as algorithm tracing and writing. Binary number and its operations and conversion are introduced thereafter, and also used as an extensive application of algorithms.

2.2. Unit 2) Programming Principles and Concepts

The follow-up unit is programming principles and concepts, which is introduced as an implementation of algorithms. In order to bridge the gap for students without any prior programming background, we choose to start the unit with turtle graphics, which have been used to teach kids basic programming concepts since the addition of Seymour Papert's 'turtle' in the Logo language in the late 1960's. Programming concepts such as loops or variables can be abstract and difficult for beginners to understand. However, blending Python with the turtle library makes these concepts more tangible as the output on the canvas allows students to see what is happening. The concept of algorithm design and pattern recognition are also reinforced as students explore drawing of more complex geometric shapes such as nested polygons. The programming concepts of sequence, selection and repetition are discussed and practiced as well. Well linked by these concepts, it is natural to introduce Python syntax, in the second half of the unit, and to apply it to perform calculations for simple and complicated sequences and series such as factorial and triangular number. This is a suitable context to illustrate programming concepts such as variables, mathematical operators and assignment statements. The programming unit is concluded with an example on the approximated value of π, using naïve version of Monte Carlo simulation, where both turtle graphics and series calculation are applied (Figure 1). This example also helps to illustrate the usefulness of computers and programming in conducting simulations related to Mathematics or Science.

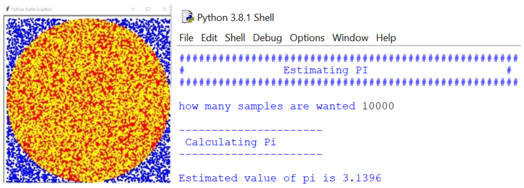


Figure 1. Screenshot of the Python program to approximate value of π.

The chapter flow from problem solving, to algorithms and finally programming is carefully chosen in order to place emphasis on thinking and problem solving, and to allow students to appreciate the connection between them.

2.3. Unit 3) Basic Data Skills

Managing and interpreting large amounts of data is essential in our digitalized world, which inspired our third and last unit. The ability to analyze, visualize and draw conclusions from large data sets is critical in the 21st century. Students are introduced to a simplified Data Science Life Cycle process, where they learn to clean, explore, analyze and visualize data with the help of spreadsheets and its related functions. The unit is designed to allow students the opportunity to apply the elements of computational thinking, process of problem solving, and to experience the process of data analysis in an authenticreal-world context. The module concludes with a last chapter on Data Security, where basic concept of cryptography and different classical cipher algorithms are discussed. The chapter also acts as a final recap on the applications of algorithms.

3. PEDAGOGY

The pedagogical approaches for this Computational Thinking module are anchored on constructivist approach to bring out engaged learning and to allow more teacher-student interaction. Students are actively involved in the process of meaning and knowledge construction via various discussions and practical sessions. We will discuss three key pedagogical approaches used in the modulehere.

3.1. Problem Solving

"Solving a problem means finding a way out of a difficulty, a way around an obstacle, attaining an aim which is not immediately attainable." (Polya, 1965: p. ix). In the problem solving unit, the 4-step approach drawn from the works of George Polya is adopted to guide students in the problem solving process (Figure 2).

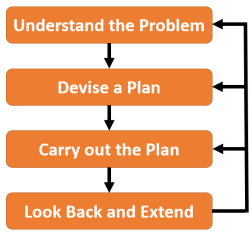


Figure 2. George Polya's 4-step approach to problem solving

Teachers first explain each step and then walk through this entire process with students in class. Students next attempt the worksheet, following the steps outlined, with guiding questions provided. The process is repeated for all questions given in the problem solving chapter.

3.2. Differentiated Learning

Differentiated instruction is about customising the teaching to cater to learners of different abilities in the same classroom. It allows higher ability learners to be stretched, without compromising the weaker learners. Differentiation is carried out in this module by setting tiered questions:

- Core standard exercises for all
- Additional basic practices for students who need reinforcement
- Optional challenging questions to stretch high ability learners.

High ability students who completed the core standard exercises ahead of the rest may proceed to attempt the challenges in class. This help to ensure all students are fully occupied and meaningfully engaged in class. Supplementary reading materials and optional contests covering topics out of syllabus are also provided for interested students to explore beyond.

3.3. Gamified Learning

The gamification of learning is an educational approach to motivate students to learn by involving game elements in learning environments. A storyline was crafted to package weekly practices into "trainings" and the larger pieces of assignments into "missions". Students enjoy their learning through the gamified learning platform where they earn certain experience points (EXP) and achieve some badges as they complete each piece of homework. They may also get to level up when they have gained enough EXP, and may even gain a spot in the leaderboard as they maintain the good work (Figure 3 and 4). To promote the habit of completing work punctually and improve their time management skills, bonus EXPs are also awarded to students who submit their work early. To stretch the better students and to encourage consistency throughout the semester, additional special badges are designed for students who are able to attain at least 80% for all tasks.

Figure 3. Example of gamified elements in the learning management system (LMS). The left shows a picture of the storyline and the right is a screenshot of a student's achievements gained, current level and total EXP.

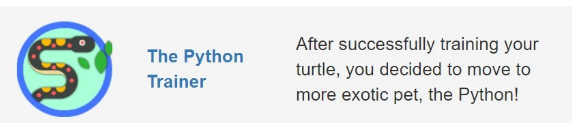


Figure 4. Each badge is designed with relevant description to supplement the storyline.

4. ASSESSMENT

4.1. Formative Assessment

There are various formats of formative assessment to facilitate learning. Pen-and-paper worksheets are issued for the problem-solving chapter. Students first attempt the worksheets individually to go through the problem solving process, followed by discussions in groups to promote collaborative learning. The worksheets will be marked by teachers with detailed feedback. Finally, students will do individual correction as teachers go through the suggested solution.

Students are given one topical lab practice weekly for each chapter. The lab practices help to monitor students' learning progress and provide ongoing feedback that can be used by teachers to improve their teaching and by students to enhance their learning. The LMS is set to provide full immediate feedback for every attempt and to allow students to have unlimited attempts until they get the answers correct. These settings promote self-directed learning and sense of excellence. It also encourages perseverance and metacognition as learners reflect on their mistakes as they attempt to debug their code until they get it correct.

In contrast, two larger scale take-home assignments are designed to allow students to apply their learnings to solve real world problems. These assignments are manually marked with feedback from teachers and bench-marked across the teaching team to ensure consistency. The first assignment is to use turtle to create an exquisite and intricate graphics following a given theme. The theme in 2020 was *"to promote safe management measures against COVID-19 in schools"*, which is carefully chosen and closely related to the student's life (Figure 5). Creative use of *for* loops, instead of hard coding, and the use of more than 8 turtle functions are encouraged. To improve students' communication skills, an explanation of the code and a description of the concept behind the graphics drawn are required in the form of a report. Aesthetics & Creativity is also part of the grading rubrics. The second assignment is on data analysis. Students are tasked to find a suitable dataset under an assigned theme (for example, education or transport), to post a few questions related to the dataset and to answer them with the findings from the data analysis and data visualisations.


```
turtle.goto(-10, 150)
turtle.pendown()
turtle.color('dark green')
turtle.begin_fill()
for count in range(9):
    for count in range(3):
        turtle.forward(13)
        turtle.left(120)
        turtle.dot(10)
    turtle.forward(6)
    turtle.right(360/9)
turtle.end_fill()
turtle.penup()
```

Figure 5. Example of graphics drawn by a student for assignment 1, with accompanying code snippets.

4.2. Summative Assessment

To meet the requirements of several learning outcomes, three different components of summative assessment are put in place. Written graded exercises are crafted to improve learners' confidence in pen-and-paper-setting, for example, algorithm reading and tracing, code reading and writing on paper, and understanding and use of spreadsheet functions. Two time-bound assessments are placed at the end of the module, namely, a lab based test which will allow them to apply their coding skills to solve problems on computers, and a final pen-and-paper examination which allows them to demonstrate their mastery of the concepts of the entire module.

5. FINDINGS

This module has received very positive feedback from students. Some common comments include fun and engaging, rich in content, breadth of coverage, and cultivating interest in computer science and problem solving. Many students also commented that from this module, they realized that Computer Science is not just about programming. On average, 77% of the cohort choose to take the follow-up programming module subsequently.

As shown in Figure 6, the gamification elements greatly motivate and engage the learners, with 93% of the students expressing they enjoy learning the materials through the gamified elements, 86% expressing the gamification motivated them to complete work and 92% expressing the feature should be used in future CS modules. The results shown in Figure 7 further supports the finding that gamification motivated students to complete their work earlier. Among all labs submitted within 7 days, the average days before submitting the weekly labs has improved from 0.29 days before deadline to 3.27 days before deadline.

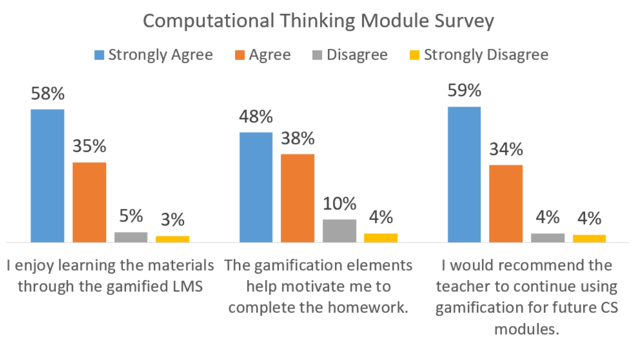


Figure 6. Results from Module Survey

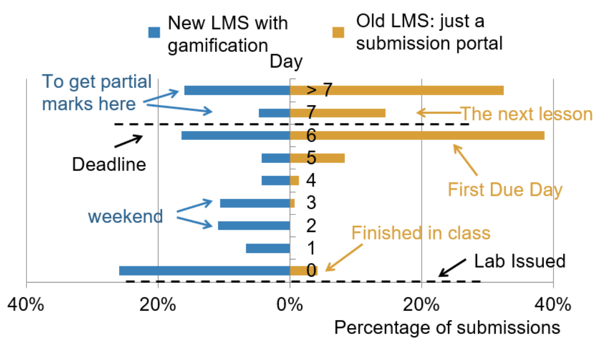


Figure 7. Percentage of submission on each day.

6. CONCLUSION

In conclusion, the Computational Thinking module exposes students to the various aspects in Computer Science and prepare them for more advanced CS modules ahead. Engaged and motivated through gamified and differentiated learning, students show an increased interest in Computer Science and obtain better learning outcomes. Through the 3 units, students are exposed to the concept of computational thinking, programming and application of problem solving in real world context through data analysis.

7. REFERENCES

British Broadcasting Corporation (2017). Introduction to Computational Thinking. Retrieved September 19, 2017, from https://www.bbc.co.uk/bitesize/guides/z4rbcj6/revision/1

Computer Science Teachers Association (2016). *K-12 Computer Science Framework*. Retrieved July 26, 2017 from https://k12cs.org/

Computing at School Working Group (2012). *Computer Science: A curriculum for schools*. Retrieved March 28, 2014 from https://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf

Goode, J. and Chapman, G. (2011). *Exploring Computer Science*. Retrieved February 25, 2014, from https://pact.sri.com/downloads/Pages-1-26-from-ExploringComputerScience-v40.pdf

Leong, W.L and Luo, Y.J. (2011). *Application of game mechanics to improve student engagement*. Retrieved March 12, 2015, from https://www.comp.nus.edu.sg/~bleong/teaching/tlhe2011-game.pdf

Papert, S. (1972). Teaching Children Thinking. Programmed Learning and Educational Technology, 9(5), 245-25

Polya, G. (1981) *Mathematical Discovery On Understanding, Learning, and Teaching Problem Solving*. Retrieved March 22, 2015, from http://www.tlu.ee/~tonu/modesimu/George_Polya_Mathematical_discovery.pdf

Polya, G. (1945/1973). *How to solve it.* Princeton, NJ: Princeton University Press.

Wing, J.M. (2007). *Computational Thinking.* Retrieved February 20, 2014, from https://www.cs.cmu.edu/afs/cs/usr/wing/www/Computational_Thinking.pdf

Wing, J.M. (2008). *Computational thinking and thinking about computing*. Retrieved February 20, 2014, from https://www.cs.cmu.edu/~wing/publications/Wing08a.pdf

Pedagogical Design of Flowcharts and Tasks to Teach Computational Thinking to Lower Secondary Students

Kester Yew Chong WONG
National Junior College, Singapore
rskester.wong@gmail.com

ABSTRACT

With the gradual ease of Computational Thinking (CT) into the Singapore mathematics curriculum, resources have been created by the Ministry of Education (MOE) Curriculum Planning and Development Division (CPDD) such as the teaching of CT using algorithms. This article seeks to draw parallels with the resources developed by MOE CPDD by re-representing the teaching of CT using flowcharts to better address the needs of lower secondary students. It also seeks to address the learning gap concerning the evaluation of students' CT ability. CT Tasks are proposed to provide timely feedback to both teachers and students on the learning of CT in the classroom. These tasks seek to provide Assessment for Learning (AfL) to allow students to evaluate their learning of CT and for teachers to review their teaching of CT and feed learning forward for future lessons. Pedagogical design of the flowcharts and tasks involving Bruner's Concrete-Representation-Abstract sequence, Vygotsky's theory of scaffolding and Tomlinson's Parallel Curriculum are discussed with illustrated examples.

KEYWORDS

flowchart, task, pedagogical design, lower secondary mathematics, assessment

1. INTRODUCTION

In 2019, it was announced in Singapore's Ministry of Communications and Information's (MCI) Workplan Seminar that a coding programme called "*Code For Fun*" will be introduced to upper primary school students in 2020 to provide them with the opportunity to learn Computational Thinking (CT) through the process of basic coding. However, it remains a question on how this learning can be continued when these students go into secondary schools in 2021 since it is not compulsory at the secondary level.

Although a new Computing subject was offered to students as an O' Level subject in 2017, only selected schools are offering it. In 2020, MOE Curriculum Planning and Development Division (CPDD) proposed the teaching of CT in the mathematics syllabus using algorithms. This helped bridge the gap of learning CT in the secondary mathematics curriculum. The four aspects of CT that were identified included abstraction, decomposition, generalization and algorithmic thinking. The mathematics curriculum provides the context for the teaching of these four aspects of CT using algorithms in pseudo-code.

In this article, we will explore what are some pedagogical theories that teachers should consider when they intend to teach CT in the mathematics classroom, and possible alternatives for the teaching and assessment of CT.

2. PEDAGOGICAL DESIGN TO TEACH CT

The leverage of pedagogy to design lessons provide a measured approach for teachers to understand the rationale and purpose of the different learning activities created for their students. By identifying specific learning objectives and relevant pedagogical theories, we seek to implement effective teaching strategies, activities, and assessments to achieve the learning of CT.

Bruner's Concrete-Representation-Abstract Sequence

Bruner proposes that children's cognitive development of concepts can be through one of these three forms: concrete, pictorial and symbolic forms (Bruner, 1966). In Singapore, it is also known as the Concrete-Pictorial- Abstract (CPA) approach. The ability of students to move from concreteness to abstractness, where there is gradual decontextualization, signifies their attainment of abstraction. The process that teachers take to teach students CT is also the enabler for students to demonstrate abstraction. CT can be introduced as a pictorial form using flowchart diagrams. While there are various representations involved in the graphical representation of computing flowcharts, we can narrow the required symbols relevant to the mathematics curriculum as shown in Table 1.

Table 1. Description of symbols used in flowchart.

Name	Function
Oval	Start or Stop of flowchart.
Parallelogram	Input or Output operations.
Rectangle	Processing arithmetic operations or assigning of variable.
Diamond	Decision making to represent the operation in which there are different flows based on conditions.
Arrow	Flow line to indicate the flow of logic by connecting symbols.
Dotted Lines	Comments to provide explanation.

In the teaching using flowcharts, it is crucial for teachers to not just focus on the flowchart itself, but on the flow that is inherent within the diagram. In addition, it will encourage greater receptivity and openness of students to learn CT by reducing the cognitive load of students. This leads to the next crucial pedagogical consideration: scaffolding.

Vygotsky's Theory of Scaffolding

Vygotsky highlights the inseparability of the learner's learning to her environment and explains how knowledge which is previously in the zone of proximal development can become the actual development level through assistance such as scaffolding (Vygotsky, 1978). Guidance to the initial learning of CT should involve mathematics content taught at the primary level, not by teaching CT using secondary level content or mentioning CT as an after-thought upon completion of teaching the secondary level content. CT needs to be taught intentionally and explicitly in the curriculum to allow scaffolding to be done meaningfully. If not, teachers will not see the relevance of using CT in the classroom as its impact on students is diminished. An example of a mathematical concept that teachers can use to scaffold students' algorithmic thinking is to recall students' primary school knowledge of expressing an improper fraction as a proper fraction, and then illustrate the process using a flowchart. This provides the opportunity for students to integrate into their schema the respective symbols involved in the flowchart. This mathematical concept does not involve decision-making statements, hence it reduces the cognitive load for learning flowcharts. Furthermore, teachers can use the concrete approach by using numbers to scaffold students towards generalization and abstraction. An example of a flowchart to illustrate this example is given in Figure 1.

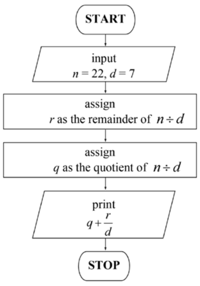
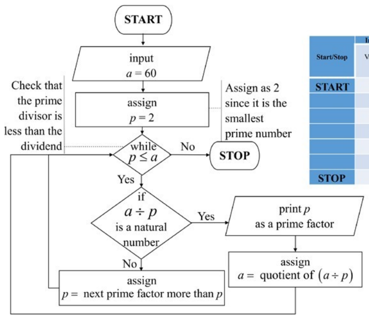


Figure 1. Flowchart of expressing 22/7 as a proper fraction.

Furthermore, teachers can scaffold the learning of CT by complementing the flowchart with a table of values. This helps to track the progression and movement of the flowchart. For example, Figure 2 shows the table of values together with the flowchart that is used to obtain the prime factorization of the number 60.



Start/Stop	Input / Assign Values		Conditions		Output
	Value of a	Value of p	while $p \le a$	if $a \div p$ is a natural number	Print
START	60	2	Yes	Yes	2
	30	2	Yes	Yes	2
	15	2	Yes	No	-
	15	3	Yes	Yes	3
	5	3	Yes	No	-
	5	5	Yes	Yes	5
STOP	1	5	No	-	-

Figure 2. Flowchart and table of values.

Assessment is vital in the teaching and learning process of students. How do teachers evaluate the learning of students' CT and sustain the engagement of students? It is vital to consider the intrinsic nature of learning to encourage the application of CT in the lives of students beyond the classroom.

Tomlinson's Parallel Curriculum

Tomlinson proposes four dimensions that runs parallel in any curriculum: the core curriculum, the curriculum of connections, the curriculum of practice and the curriculum of identity (Tomlinson et. al., 2006). The richness that CT can bring to students will be enhance if all four dimensions to the parallel curriculum is included. In the teaching of CT using flowcharts, the process that is used by students to solve questions should to that which is illustrated in the flowchart. It is better to have a flowchart that is not as efficient, than to have a flowchart that confuses and deters students from appreciating the CT involved. Students need to relate CT with the Core Curriculum.

In the assessment of learning for CT, teachers can involve the curriculum of connection to help students relate how CT is used in their daily life. For example, all secondary students can sort numbers in ascending or descending order observe the use of it on their phones in different applications. Teachers can relate to students' experience and lead them to question the thinking involved in the sorting of numbers by creating a CT Task. An example of a CT Task used to assess the learning of CT is given in Figure 3.

Task 1: Sorting of Five Numbers

1. Label the number '5' on a paper cup. Ensure that the number is clearly written and can be seen from a distance.
2. Describe the process of sorting the following numbers in ascending order.

$$7, 6, 9, 8, 3$$

Video-record your description, showing step-by-step the process involved by illustrating it clearly using the paper cups.

Task 2: General Algorithm of Sorting Numbers

3. By considering the process used in **Task 1**, describe an algorithm that can be used to sort any given list of numbers. Present your algorithm as a flowchart using PowerPoint.
4. Validate your flowchart by testing the correctness of the algorithm by sorting the following numbers in ascending order.

$$-5, 0, -1, 5, -7, 1$$

Video-record the description of sorting the following numbers in ascending order by illustrating it clearly using the flowchart and a table of values that follows the sorting process.

Figure 3. CT Task to assess the learning of CT.

Teachers can differentiate the learning of students by extending the learning of students who are ready to apply CT to more complex contexts such as applying it to do mathematical modelling (Ang, 2020). This encourages students to relate CT as a curriculum of practice and identity.

3. CONCLUSION

The pedagogical designs, together with the illustrated examples, aim to provide perspectives for teachers to bring CT to live in their mathematics classroom. While challenges may exist in lesson enactment, students with a keen sense of CT can have a deeper appreciation and better grasp of the technological-driven world they are in. Future work can be done to study the effectiveness of teaching CT in practice and also the usefulness of CT Tasks to evaluate students' learning of CT.

4. REFERENCES

Ang, K.C. (2020 Dec). Computational thinking as habits of mind for mathematical modelling. In Yang, W.C. & Meade, D. (Eds.), Proceedings of the 25th Asian Technology Conference in Mathematics (pp. 126-137). Mathematics and Technology, LL

Bruner, J. S. (1966). Toward a theory of instruction. MA: Harvard University press.

Tomlinson, C. A., Kaplan, S. N., Purcell, J., Leppien, J., Burns, D. E., & Strickland, C. A., (2006a). The parallel curriculum in the classroom, Book 1: Essays for application across the content areas, K–12. Thousand Oaks, CA: Corwin.

Vygotsky, L. S. (1978). Mind in society: the development of higher psychological processes (M. Cole, V. John-Steiner, S. Scribner, E. Souberman, Trans.). Cambridge, MA: Harvard University press.

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Rethinking Computational Thinking Implementation in K-12 and Challenges Faced

Susanna SUNIL
Global Public School
India
susannasunil@globalpublicschool.org

ABSTRACT

The school has adopted the computational thinking curriculum since 2017 to equip students with logical reasoning and problem solving skills. The teachers have worked on ways to improve the efficacy of the curriculum as a whole - not just in terms of content but also in terms of stakeholder feedback. This paper describes the integration of Computational Thinking curriculum in phases across Grades 1 to 8, the implementation and challenges faced.

KEYWORDS

CTImplementation, Computational Thinking in K-12, Challenges Faced in CT, CTEducation

1. INTRODUCTION AND BACKGROUND

Global Public School is a K-12 day boarding cum residential school with 1750+ students in Kochi, Kerala, India affiliated to the national education board, CBSE. Established in 2006 with an equal ratio of boys vs girls the school has a mix of students from high income, middle class, and a small percentage from the lower end of the spectrum. Students come from both - business as well as salaried backgrounds with a sizable percentage from NRI(Non-Resident Indian) families in UAE, US, UK & Africa. The school also has students with special needs.

The school has adopted the CSpathshala Computational Thinking curriculum since the end of the 2016-17 academic year as computational thinking is wider in scope and involves understanding a problem, designing a solution and expressing it in a form that a human or a machine can execute.

CSpathshala (www.cspathshala.org) is an Association for Computing Machinery (ACM) India education initiative to bring a modern computing curriculum to Indian schools. CSpathshala has developed a class room, activity based Computational Thinking Curriculum for K-12. CSpathshala has adopted the unplugged activity-based approach to teach Computational Thinking (CT) without the use of computers.

2. ADOPTION OF COMPUTATIONAL THINKING CURRICULUM

As schools in India do not have a formal prescribed curriculum for primary and middle school, the decision of choosing a curriculum can be taken by the school management. As this is a co-scholastic subject, the focus is on enhancing the skill of the students and assessment is conducted through activities.

The school's Information Technology (IT) team have regularly reworked the ICT content to ensure that age appropriate skills were imparted to students. The objective of this team has been to adopt techniques that will keep each student abreast of the latest technologies as well as to learn to learn.

The team reviewed various curricula but found that most standard curricula were restrictive, focused on digital literacy and dependent on software versions which required frequent hardware upgrades. In 2013, the team developed a milestone-based in-house curriculum that was independent of software version/ company and hardware shortfalls and focused on getting students to understand the power of using computers as a tool in every subject learnt. We taught Word processing, Spreadsheet and Presentation software in generic terms and made students experience these in GSuite, Microsoft Office and Open office. Students were also taught Scratch programming.

Although the school was teaching computers, it focused primarily on digital literacy and a bit of programming. Then in 2016-17, the CSpathshala team came to the city, and the teachers were introduced to Computational Thinking and we found a kindred spirit. The main objective of the CSpathshala introductory workshop was to motivate the teachers to teach computing as a science through enhanced understanding on why Computer Science should be taught in schools and how it is different from ICT.

CSpathshala national curriculum committee comprising computer scientists from top academic institutes as well as IT industry experts developed a computing curriculum suitable for K-12. Here was a curriculum with a step by step approach to Computing rather than "Computers". Computer Science is the study of the principles and use of computers.

Students learn to use the computers as tools with ease, but not much beyond. Computational thinking helps in training the mind to think logically, from understanding the problem, breaking it down to simple solvable steps, finding out relevant and irrelevant things, making connections, looking for patterns, making generalizations, devising algorithms, representing things such that others understand it(including computer, through coding).

We also found that this curriculum had integrated CT thematic concepts seamlessly using daily life examples and providing students an opportunity to learn with fun. As CSpathshala has a readymade curriculum, syllabus and content, the teachers could easily adapt it and take it to the classroom.

3. PHASES OF IMPLEMENTATION

The school has five dedicated IT teachers all with a computing background. The school timetable has one period (40 minutes) for Grades 1 to 5 and two periods (90 minutes) for Grade 6 to 10 dedicated per week for teaching Computers, both for digital literacy and for CT.

Year 1: At the end of 2016-17, our teachers attended the introductory training on Computational Thinking by the CSpathshala team. It helped them understand CT and the interesting activities requiring problem solving and computational thinking skills. The training enthused us so much, that we conducted a pilot with selected lessons for Grade 2 to 5.

We call this Phase 1 of implementation.

Year 2: The encouraging feedback from the teachers and students led to taking a radical call to replace the entire IT curriculum of Grade 1 to 8 in the academic year 2017-18 to CSpathshala curriculum. This had meant a huge paradigm shift for the teachers and to equip the teachers on CT, CSpathshala team conducted a teachers training program to train them on problem solving and computational thinking skills. The training program focuses on topics such as, decomposition, patterns, abstraction, algorithms, analysis and programming with day to day examples. The team also demonstrated the interesting activities requiring problem solving and computational thinking skill to equip the teachers.

We call this the Phase 2 of implementation.

Year 3 and 4: As CSpathshala follows the unplugged approach, the activities were all pen and paper. In 2018-19, we received feedback from both teachers and parents on how Computer periods should have hands-on sessions in the lab. In 2019-20, the introduction of AI in grade 8 by our affiliated Board of Studies, and the need to include Robotics, found us reworking our annual plans and a drastic reduction in CSpathshala content.

We call this Phase 3 of implementation.

In 2018, CSpathshala conducted the first Bebras India Challenge, and we conducted the challenge for students from Grade 6. Bebras (www.bebras.org) is an international student Computational Thinking Challenge organised in over 60 countries and designed to get students all over the world excited about computing. In 2019 1,000+ students from Grades 4 to 12 participated in the challenge.

Year 5: Owing to the pandemic, in 2020-21, there was no implementation of CT curriculum. The school conducted the online 2020 Bebras India Challenge across Grades 3 to 12.

4. PERIODIC REVIEWS AND CHANGES IN CURRICULUM

Working towards the goal of developing problem solving capabilities, students were encouraged to ask questions, identify the problem, find the pattern and then connect the dots in different scenarios. The teachers conducted various activities linking them to the thematic areas of computational thinking. So to bind these seemingly disconnected processes, students were able to see and understand that complex problems can be simplified, using decomposition.

During Phase 2, after the first term, the teachers shared that the students found learning new concepts of systematic counting, patterns and puzzles like Sudoku interesting, but were restless as they were used to hands-on computer activities. The change in curriculum drastically reduced their time in the lab and students would complain to the teacher. A similar feedback was shared by parents that Computer classes were all about theory and there were not enough computer applications being taught.

To address these challenges, in 2018-19, during phase 3, the IT team reviewed the curriculum and did a course correction, reworked our Annual plans integrating CT activities with tools like word processors and spreadsheets. W e also had a review meeting with the CSpathshala team who conducted student sessions to demonstrate and train teachers on the approach to be used in the classroom using CSpathshala resources.

Worksheets were taken up as a lab activity to address the problem of computer period being equated to practicals in the lab. Some of the application based changes that we introduced are mapping of the activities with suitable tools like PAINT, Word processor and Spreadsheets. Some illustrative examples are mentioned below:

- Using PAINT for concepts like patterns in shapes and coloring, Rangoli patterns (traditional Indian art), creating symbols of their own
- Using WORD PROCESSOR for drawing a table using Insert Table and shading the first alphabet of their name using colours.
- Using Spreadsheets for creating Sudoku and solving it, and learning to process information to SORT and FILTER i.e., alphabetical order, highest to lowest, and vice versa.

In 2019-20, taking into consideration new initiatives of CBSE, and consistent regular feedback from parents and students, CSpathshala content was reduced in Grades 6 to 8 with a view to strengthening foundational skills in Grade 1 and 2.

5. TEACHERS FEEDBACK AND LEARNING PRACTICES

Teachers shared that the teaching aids of CSpathshala were effective tools to invoke an interest in young minds and also helped develop a systematic approach to problem solving through the activities:

Patterns: One of the thematic areas which is most popular amongst both students and teachers is the concept of patterns. In the classroom for Grade 2 we integrated patterns in shapes, by exploring artwork from Piet Mondrian, a famous painter and the impact was that the students recreated the masterpieces using Paint. The outcome of these activities was increased participation with students identifying patterns in diverse areas all around them. Students have come up to teachers with examples of patterns.

(Link to video: https://www.facebook.com/GlobalPublicSchoolKochi/posts/1748415075214689)

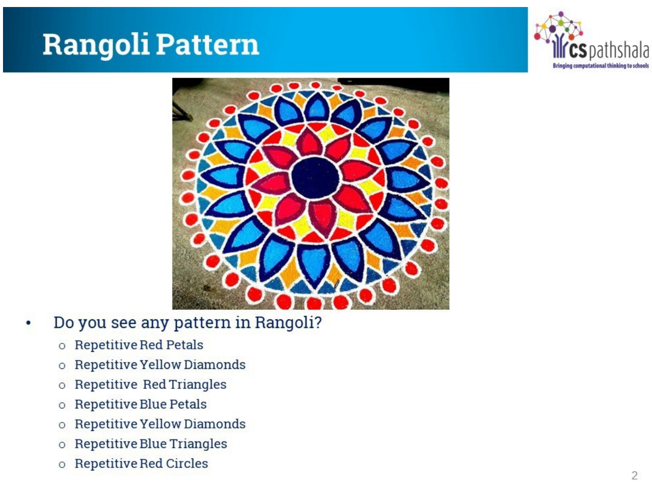


*Figure 1.*Example of Patterns (Rangoli) slide

Algorithms: Happy Maps is a lesson with a goal to teach students to reach a destination using directions and follow simple instructions and through these activities develop algorithmic thinking. Students were taught through simple rhyme and movement about the concept of up, down, left, right, forward, backward and also on movement. The end result was that even Grade 1 students were able to create a simple algorithm for reaching a destination.

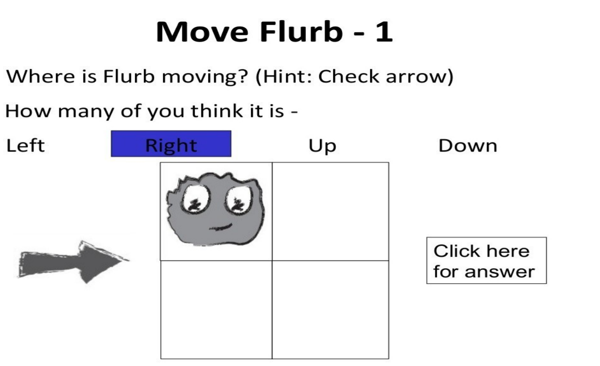


*Figure 2.*Example of Happy Maps slide

CSpathshala lesson plan on Arranging/Sorting data gave many examples from daily life which did the magic of increased participation of students. One of their favourite activities was "How to locate your shoes". The students came up with multiple solutions to the solve problem and also connected this with similar problems (generalisation).

Systematic Counting: Binary Numbers, a concept which is generally confusing to students of Grade 5, became a fun activity using the CSpathshala teaching aids.

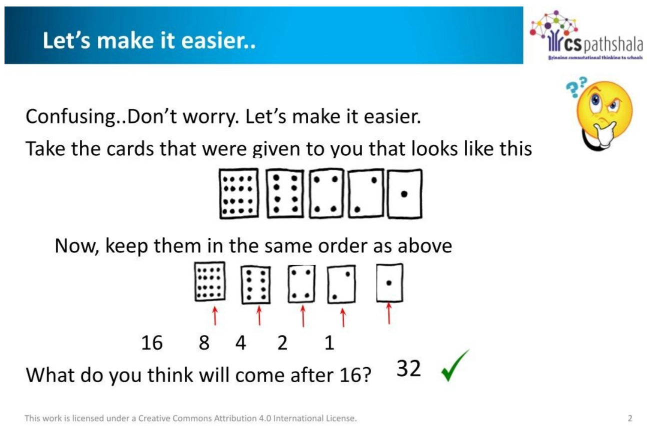


*Figure 3.*Example of Flash card slide (Binary/Decimal)

The teacher introduced the binary number system to students demonstrating the decimal number system as bundles of ten cards, place values as units tens and hundreds and binary as bundles of 2 place values as units. Counting in binary was easy for the students as the teacher demonstrated the activity using the flash cards provided by CSpathshala. Systematic counting using activities was a new concept and they were able to move to binary counting with ease.

Data: Sharing the Grade 3 teacher's feedback on arranging and analysing data: "Earlier it was a challenge to teach students concepts in databases. Using the CSpathshala lesson plan which used a small number of entities (decomposition), made it easier for the students to understand the need to arrange data. The students responded well as they could visualise the data and were able to answer questions without any difficulty".

Alignment of CT activities with Mathematics: As Math Teachers were not aware of CT activities related to Math concepts, we organised a session for them to integrate and align some of the activities as part of in Grade 1 and 2. These activities led to **reinforcement of mathematical concepts through the Computational Thinking activities**.

Story Telling: Across Grade 1 to 5, teachers have shared that storytelling eased the heaviness of content. Students across the board were introduced to IT personalities with a unique story of their programming journey. So Tic-Tac-Toe was introduced and played with a background introduction of the first programming experience of 13-year-old Bill Gates. The storytelling strategy created interest among children to connect to the concept and motivated students to develop a step by step approach and winning strategy through this activity.

Bebras Challenge: The school has been conducting the Bebras Challenge for the last 3 years and students have shared encouraging feedback. A Grade 5 student opined, "The Bebras challenge has thinking questions and it was fun. I understood a lot of things from the questions. The Questions were mind puzzling questions. And I really loved it. Hope more like these will come".

6. WAY FORWARD

Adoption of CSpathshala has been a huge learning experience both for the teacher community and the taught. As we move into our sixth year of implementation we are a bit more circumspect and wiser and have gained insight on implementation strategies and the curriculum is a mix of CSpathshala and our own in house program

The National Education Policy 2020 outlines the vision of India's new education system. This policy decision has invoked a new interest and also led to easier acceptance of computational thinking amongst all stakeholders.

NEP 2020 Section 4.25 mentions that:

It is recognized that mathematics and mathematical thinking will be very important for India's future and India's leadership role in the numerous upcoming fields and professions that will involve artificial intelligence, machine learning, and data science, etc. Thus, mathematics and computational thinking will be given increased emphasis throughout the school years, starting with the foundational stage, through a variety of innovative methods, including the regular use of puzzles and games that make mathematical thinking more enjoyable and engaging. Activities involving coding will be introduced in Middle Stage

The National Education Policy (NEP) 2020 in clause 4.25 states that

Thus, mathematics and computational thinking will be given increased emphasis throughout the school years, starting with the foundational stage, through a variety of innovative methods, including the regular use of puzzles and games that make mathematical thinking more enjoyable and engaging. Activities involving coding will be introduced in the Middle Stage.

We feel justified that our early forays into computational thinking has helped shore up valuable expertise amongst our teachers in ensuring the implementation of the NEP 2020 will happen smoothly. The revised Computational Thinking curriculum from CSpathshala which we have adapted within our curriculum for 2021-22, will make a big difference in the forthcoming academic year

7. CONCLUSION

One of the challenges faced by educators is that the existing school curriculum is too crowded to add a new subject, the time allotted for ICT across Grades 1 to 8 was restructured to teach computational thinking side by side with computer literacy. Also the well-structured CSpathshala curriculum and syllabus documents along with the teaching aids provided a clear roadmap for adoption of CT.

Our experience of onboarding a computational thinking curriculum over these past years, has underlined the importance of teaching CT in every grade. It has also become amply clear that teacher development is the key to implementing this change effectively.

With the implementation of the computational thinking curriculum with CSpathshala, we are sure that the students will be the harbingers of change in the future and the strong problem solving and critical thinking skills that they develop will influence every varied field of work that they take on in the future.

8. BIBLIOGRAPHY/REFERENCES

Computational Thinking in K-12 Education: **https://CSpathshala.org/2017/10/25/computational-thinking-curriculum/**

Computer Science Unplugged csunplugged.org/

CSpathshala syllabus and CSpathshala curriculum documents: https://CSpathshala.org/curriculum/

National Education Policy 2020 document - National Education Policy 2020

Teaching aids Resources from www.CSpathshala.org

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Integration of Computational Thinking in Upper Primary (Grade 6-8) Math in Tamil Nadu, India

Malarvizhi PANDIAN[1], Krithika KRISHNAMOORTHY[2*]
[1]Chennai High School, Strahans Road, Tamilnadu, India
[2]Azim Premji University, Bengaluru, India
vrpmalar16@gmail.com, krithika.apf@gmail.com

ABSTRACT

Computational Thinking (CT) curriculum has been introduced as part of Mathematics in the formal state government curriculum since the academic year 2018-19 across Grades 1-8 in nearly 30,000 schools within Tamil Nadu, a state in India. This paper focuses on the introduction of the Computational Thinking component in Grades 6-8. It describes the process of integrating CT components in Math curriculum, especially the transformation from "Data Handling" to "Information Processing" with a focus on problem solving and data organization part of the curriculum in computational thinking. It describes implementation and challenges faced.

KEYWORDS

CT in Mathematics Education, CT in K-12, Information Processing, Math in digital era, Government Schools.

1. INTRODUCTION

The goal of mathematics education is not to provide computational skills (that calculators can accomplish better) or informational knowledge (that search engines can deliver easily), but to influence citizens' thought processes in such a way that society can manage its resources efficiently and equitably. How is this to be done? The content areas of mathematics provide plenty of opportunity for the child to train the mind to think logically, abstractly, critically and creatively.[8]

In India, Tamil Nadu state government has taken a lead to implement the integration of computational thinking in, mathematics at the elementary stage, across all the government schools, in a phased manner from grades 1 - 8 as they valued CT to be the dire need of the digital era: This was achieved in two phases: Phase1 in academic year 2018-2019 for grades 1 and 6 and Phase2 in academic year 2019-2020 for grades 2-5 and 7-8. Later in 2020, the new National Education Policy (NEP) also emphasized on inclusion of CT in math.

The framework for mathematics curriculum had been provided in a position paper [8], and the syllabus was mainly based on it. Further, during the process of revamping syllabus and textbooks, the resource group referred to the following resources: National curriculum Framework (2005), existing mathematics textbooks and syllabus of NCERT (India), Kerala (India) and Singapore. Teacher Community & Teacher Educators such as Lecturers of DIET (District Institute of Education and Training), Mathematics Professors from institutions of higher education, educational functionaries of the Government and members of non-government organizations were involved in this process, organized by TNSCERT (Tamilnadu State Council of Educational Research & Training).

2. CHANGE FROM EXISTING CURRICULUM WITH INTRODUCTION OF CT

The existing syllabus had data handling, in which the importance was given mainly for collecting / creating data and for simple visual representation and to some extent for interpretation. The position paper on Mathematics in the Tamil Nadu Curriculum Framework 2017 states: *"It is almost a cliche to talk of the ubiquitousness of computers and Internet in modern life. Algorithms are taking over the running of many aspects of everyday life of the citizen, and understanding the world is going to increasingly involve understanding of its digital manifestations. Moreover a strong foundation for computational thinking will be essential for children growing up in this century. As it happens, such understanding and thinking lies squarely within the realm of mathematics in school."*[8]

Based on the position paper, the existing curriculum, syllabus and textbooks have been revised and updated for Mathematics by replacing content on "Data Handling" with "Information Processing" for classes 1 to 8, to bring computational thinking into school mathematics.

"Students should explore different methods of arranging, organizing, analyzing, transforming, and communicating information, and understand how these methods are used for information processing."[5]

Information processing has extended the scope of data handling, through various activities to explore and understand the depth of the data concept such as data organization, data representation, data analysis, pattern recognition, looking for connection/ abstraction, describing processes, making predictions, making decisions and so on , towards mathematical thinking through CT. So, the Information processing units provided scope to explore the depth of data modelling, through schematic problems and activities, which are discussed in this paper.

The major challenge was to translate such an understanding of CT into learning units for children, keeping in mind their developmental needs as well as low resource situations in which our schools are situated.

3. ALIGNMENT OF THE SYLLABUS AND CREATION OF TEXTBOOKS CONTENT OF UPPER PRIMARY MATH (GRADE 6-8)

"The goal of teaching is to design and provide experiences that facilitate the construction of knowledge"[2], which is what constructivism emphasizes on. It is not that children have never used CT in the real world earlier or that it is only after the introduction of the CT component that children are going to learn and use it in life for the first time. Rather, there are many activities that not only children but all of us use in daily life that have CT components.

"For instance, someone who has completed schooling and drives an auto rickshaw for a living, should be able to speak of what he earns on average, reflect on variations in income, consider what changes would be needed on a daily basis if the monthly income were to increase by half, and discuss the relative desirability of long distance and short distance rides. All of this involves some calculational ability for sure, but in the absence of mathematical thought, the driver would never make the calculational effort at all, and would very likely be unable to take charge of his practice in a professional manner. A similar remark would apply to a majority of the millions of self-employed in the country."[8]

To link these daily life experiences with the CT component and aligning to the goals of the curriculum/education, the resource group worked on a systematic change in the syllabus and the textbooks content across the following thematic areas, as presented in [8]. The structuring of CT into these components follows the design of the CSPathshala Curriculum [5].

Systematic listing, Counting and Reasoning: Hands-on activities in mathematics give students opportunities not only for learning number, shape and quantity, but also in counting, arranging, organizing and reasoning. For Example: In Grade 8, to teach systematic listing and counting, the teachers use the example of "Praveen's Dresses", with 3 shirts, 2 pairs of jeans and 3 pairs of shoes. Using such daily life examples that students can relate to, has led them to explore how many possible ways one can vary the combination of choices.

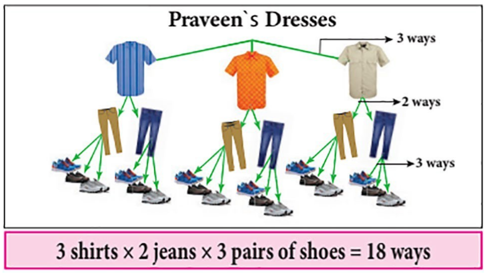


Figure 1. Example of Systematic Listing, Counting & Reasoning

This data organization connects with the concept of multiplication (say cartesian product).

Educational implications: The children were asked to describe the process, which in turn paved the way for using the math language and also trains the mind in arranging in different ways. By the use of tetrominoes, the children see the connections between the shapes, orientations and visualizations. While exploring things and describing the process, there is freedom of thought, exploration and expression that in turn reduces fear of math and opens the door for unlearning certain things since data handling is only limited to visualizations and interpretations, whereas these activities are used also for multiplication purposes and learning new things.

Modelling: Understanding the existing model and exploring different ways of remodeling based on different criteria.

Example-1: The students are given a task to change the school time table of the class and are given a set of constraints .

Example-2: For a class 6 student, she/he knows the different ways to reach her school and choose any one path to go to the school. When the student is getting late, or considering a variation in the mode of commutation or considering a safe route during the rainy season, they implicitly choose the best path without having known the process of exploring different paths. Only when the teacher explains the concept of data modelling through a daily life example, students learn the process involved in choosing the shortest path or cost effective path or efficient path or feasible/safe path, that involves comparing paths, selecting/eliminating based on criteria, comparing with other concepts like time, distance etc...through CT component. In this way, CT helps in mathematization.

Educational Implications: Student understands the current form, interprets the data, understands the required change in criteria, looks for connections, does some relevant selections and some elimination of data (irrelevant), creates or rearranges the existing form to the required form.

Patterns, Iterative Patterns and Processes: Iterative patterns and processes involve repeating a single step or sequence of steps many times[5]. In nature, certain things are repeated in different time, space, shape, colour, sound, movements etc. In the textbooks for primary classes, we had exercises involving math, CT, art and design thinking by introducing patterns in sounds, body movements, shapes, colours and different combinations of them. Let's see an example from Upper primary classroom transaction: The students identified iterative patterns in the time, day/night, days of the week, month of the year and seasons. Using these as foundation, teachers used the patterns in fruits and vegetables, patterns observed in honey combs and flower petals (1,3,5…) and linked these to explain Fibonacci series. Integrating CT approach in connecting various concepts through patterns, helped teachers with abstract concepts like arithmetic progression, euclidean algorithm, etc.

Educational implications: The children see aesthetic aspects of math through patterns and also predict long term behaviour based on the observations made. They find math around them by connecting various things from nature to patterns to various other concepts in math to making predictions in real life.

Following and Devising Algorithms: Breaking down the solutions into small simple steps to make any follower of the algorithm to reach the solution. It could be coding if the executor of the algorithm is a computer or it could be simple instructions if the executor of the algorithm is a human.

Example: In Class 8, there is an activity for the children to prepare a shopping list based on a given budget, for the list of items and on different constraints. In some schools teachers also conducted this as a field activity, taking the students to the market area to compare the prices of the items in their shopping list with the real prices in the market, with the wholesaler and in the supermarket. The children noted down the prices, compared the differences in prices, found the cost effective strategy and looked for optimized purchase.

Figure 2. Example of Following and Devising Algorithm

Educational implications: Children could break down complex tasks into manageable or workable tasks, find out similarities, dissimilarities, make connections (in this case aligning with the criteria), make some eliminations, some selections, zero down to decisions, generalize things, that in turn help them in devising algorithms based on different constraints.

According to National Curriculum Framework 2005, *making room for processes such as visualisation, use of heuristics, estimation and approximation, optimization, use of patterns, use of multiple representations, reasoning and proof, making connections, mathematical communication, and so on constitutes the difference between "doing mathematics and swallowing mathematics, between mathematization of thinking and memorizing formulas, between trivial mathematics and important mathematics, between working towards the narrow aims and addressing the higher aims".* [8]. This resonates with the approach used in creating CT content.

3. SHIFT IN PEDAGOGICAL PRACTICES

Traditionally Maths has been taught by teachers using the black board and chalk with very few hands-on activities for children. Also, teachers perceive solving problems as mechanical tasks, which leads to rote learning without an emphasis on connecting to concepts. The textbooks also often lack content for teachers on activities that could be used in the classroom.

Teachers who have now used the CT components have experienced in the classroom,as the Information processing lessons enabled both teachers and students to explore patterns around them in different combinations of sound and body movements etc. The children in turn predicted what step may come next for what kind of rhythm. Through various activities in the textbook children were given space to enjoy, to explore, to approach heuristically by understanding and creating patterns using combination of colours and shapes and combination of sound and movements. Children were allowed to sing, make sounds, dance and in fact, learn using their own body movements, which one can't observe in normal classrooms.

Traditionally math was taught as a bunch of tricks (for example, to compute HCF or to find square roots). The integration of CT into math could give a chance for the students and teachers to reflect on the processes behind these tricks, their correctness and efficiency.

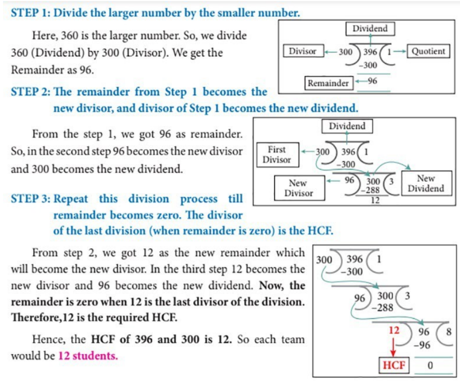


Figure 3. Example of Shift in Pedagogical Practices

"Emphasis on procedure without accompanying understanding can translate to computational ritual and fragile learning that cannot withstand even small changes."[8]

S.Subramaniam, Panchayat Union middle school, Udaiyalipatti, Pudukkottai District, and teaches for grades 6-8 shared, "Before the information processing unit was introduced, we used only a blackboard to teach the concepts. Out of our own interest, we used to create some activities for the children. But, now after information processing was integrated with math, the textbook itself has suggested a lot of activities, like tetrominoes, data organizations, for finding out all possible combinations and the like. It is very easy for us. The children also do a lot of activities to understand clearly and deeply much beyond conceptual level. It doesn't end with this class, but these experiences help them even when they have to appear for some competitive exams, which demands critical thinking."

The new content in the unit has activities along with instructions on conducting the activities and connecting them to not only CT concepts but also connecting and reinforcing concepts from other subjects.

Kalpana, Government school teacher, Alambakkam, Trichy, expressed her views on Information practices as follows: "Students started going out to explore things, to look for data, collect data, organize, interpret and even started to predict things based on the observed data/patterns. A step ahead, the children's mind has shifted from fear of problems to creating problems. Some of those problems were also added in the state level question bank booklet. Also due to covid-19, the state had decided to reduce the content. We have reduced the content from other units but in the information processing unit, we have just reduced the number of examples, but the content and exercises are left as it is, as we could find value in it."

4. CHALLENGES

Teachers Approach towards CT in the Classroom: We know that ultimately it is the teachers who are the real change makers, who have the sole responsibility of connecting the real life experiences and the curricular goals. TamilNadu government preferred the term Information Processing for this entire CT track, as "math for digital era". So, computational Thinking was integrated with math, without even using the term CT.

Another concern is mishandling of the concept. What do we mean by mishandling? How do teachers define and perceive CT? It may be limited to coding, digital literacy and Information and Computer Technology. As teachers may have a narrow perception, the real purpose of CT may not reach the children.

For effective teaching of CT, it is important to engage with the teachers and provide them with handholding support. Lack of resources for building teachers capacities for CT is a challenge.

For those students who are interested in coding, there is an option in the high school curriculum to take up programming courses. As CT provides foundational skills, the children develop the ability to solve problems systematically and hence coding becomes easier for them, which is just one of the outcomes of the processes as mentioned in CSpathshala curriculum as:

”We want to convey that computing concepts and fundamentals do not depend on particular technology or software or programming languages. Technology has a short shelf life and will serve our children for the next few years, whereas, fundamentals will stay with them for several decades.” [5]

Assessment: As CT in math itself is a new experience for the teachers themselves, there is a lack of teacher preparation and tools to assess CT skills. CT has been introduced through many activities that encourage children to explore, to make connections, to describe the process, to change representations, to discuss things and so on to enhance their problem solving skills. Hence, the assessment of CT skills is a challenge. According to the Math position paper, it is suggested that “*all assessment in mathematics should move towards becoming meaningful problem solving opportunities that enhance learning. It recommends:*

- *At every level, a small set of problems should be challenging and non-routine, calling for making connections and combining concepts.*
- *Over the years, we should gradually introduce and increase assessment of process skills such as the ability to visualize, to abstract, to change representations, to search for counterexamples, to provide arguments etc.”*[8]

5. CONCLUSION

“Classroom processes need to improve the child’s ability to mathematically articulate, analyze and solve meaningful problems. Textbooks and other educational material need to enhance the child’s ability to make rich connections across mathematical ideas.”[8]

This is just the beginning of the first step in integrating CT in upper primary math towards a long journey of making CT in math meaningful and enjoyable for the children. What we wish to emphasize is that such reasoning underlies the science of data and information organization, and getting our children tuned to such thinking at once expands their mathematical abilities and prepares them better for the digital era. This is important for our children to eventually contribute to the information revolution and not grow up only as its consumers.

6. BIBLIOGRAPHY/REFERENCES

Buffalo University, on “Constructivism”: http://www.buffalo.edu/ubcei/enhance/learning/constructivism.html#:~:text=Learning%20is%20inherently%20a%20social,facilitate%20the%20construction%20of%20knowledge.

CAS-UK. Computing at School Working Group http://www.computingatschool.org.uk

Computer Science Unplugged: csunplugged.org/CSPathshala: https://cspathshala.org/curriculum/

Freudenthal, H. Mathematics as an educational task, Dordrecht: D. Reidel Pub., 1973. “The goals of Mathematical education”, in ComMuniCator, the magazine of the California Mathematics Council, 1969.

Position Paper on Mathematics, Tamil Nadu Curricular Framework 2017, Available from State Council of Education Research and Training Tamil Nadu.

Position paper on teaching of mathematics by the national focus group, National curriculum framework – 2005, National council for education, research and training, 2006. (PDF) Position Paper on National Focus Groupon Teaching of Mathematics

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). A Model Curriculum for K--12 Computer Science. ACM/Association for Computing Machinery.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Computational Thinking and Teacher Development

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Bebras Challenge and PANDAI Movement Introducing Computational Thinking To K-12 Teachers in Indonesia

Adi MULYANTO[1], Irya WISNUBHADRA[2*], Inggriani LIEM[3]
[1]School of Electrical Engineering and Informatics ITB, Indonesia
[2]Atma Jaya Yogyakarta University, Indonesia
[3]Bebras Indonesia
adi@std.stei.itb.ac.id, irya.wisnubhadra@uajy.ac.id, inge@informatika.org

ABSTRACT

In this paper, we present Bebras Challenge as an enabler for Computational Thinking introduction to teachers, and "Gerakan PANDAI", a project supported by google charity organization, for disseminating Computational Thinking to 22.000 teachers in 22 cities of Indonesia during 2020 - 2021. The deployment strategy is by creating a network of mentors, volunteers from universities. The delivery of the program has been adapted to online distance learning due to the COVID-19 pandemic situation. One of the important remarks is that teachers need support from universities in our country since they have many students and have no time for research and self-development.

KEYWORDS

Computational Thinking, Bebras challenge, scalable project, teacher training

1. INTRODUCTION

In this Industrial Revolution 4.0 and Society 5.0 era, Computational Thinking (CT) is the new literacy in the 21st century, such as reading, writing, and arithmetic (Wing, 2012). Therefore, CT needs to be introduced to children from an early age. However, considering its nature, we cannot have CT as a subject matter. It should be infused into the existing subject area as an aspect.

In Indonesia, most of the teachers are not familiar with CT, so they need training in order to be able to infuse CT in the student's learning process. The Bebras Computational Thinking Challenge is a way to bring CT to schools, and it was proven by our first experience with dozens of schools. But then the question arises: "How can we make a scalable impact so that CT can be introduced to the whole country where the result of the PISA Test was not good?"

2. CURRENT CONDITION

Indonesia is a very large archipelago with more than 270 million population. Indonesia has more than 25 million K-12 students studying in elementary, middle, high school, and vocational schools. The schools consist of public schools, private schools, and schools managed by the Ministry of Religion, that area Islamic education-based schools called madrasah.

Indonesia's school participation rate is quite good and continues to increase, but the quality of education still needs to be improved (OECD, 2018). The PISA (Programme for International Student Assessment) is a triennial survey of 15-year-old students that assesses the extent to which they have acquired the key knowledge and skills essential for full participation in society. The assessment focuses on proficiency in reading, mathematics, science, and an innovative domain. Students in Indonesia scored lower than the OECD average in reading, mathematics, and science. Compared to the OECD average, a smaller proportion of students in Indonesia performed at the highest levels of proficiency (Level 5 or 6) in at least one subject; at the same time, a smaller proportion of students achieved a minimum level of proficiency (Level 2 or higher) in at least one subject. This result showed that we have to improve the students' High Order thinking ability. Some 88% of students in Indonesia (OECD average: 74%) agreed or strongly agreed that their teacher shows enjoyment in teaching. In most countries and economies, students scored higher in reading when they perceived their teacher as more enthusiastic, especially when students said their teachers are interested in the subject.

3. INDONESIAN BEBRAS CHALLENGE

Bebras is an international initiative aiming to promote Informatics (Computer Science or Computing) and computational thinking among school students of all ages. Participants are usually supervised by teachers who may integrate the Bebras challenge in their teaching activities (Bebras, n.d.). The challenge is performed at schools using computers or mobile devices.

As part of IOI country leaders communities, we started to know the Bebras Community from the founder of Bebras, who is also one of the IOI International Committee. In 2016, Indonesia joined the Bebras community as an observer at Bebras International 2016 workshop in Bodrum, Turkey. In the 2017 workshop, Indonesia has been accepted as a Bebras Community member.

In the 2016 Bebras Week, Indonesia started introducing The first Indonesian Bebras challenges for upper-elementary school (K4-K6), junior high school (K7-K9), and high school (K-10-K12) students. Indonesian Bebras participants continued to increase, from 1680 (in 2016) to 16186 (in 2020). In 2020, an additional Challenge was introduced, for the younger kids (K1-K3) Figure 1. shows the increase in Bebras Challenge participants..

The students who participated came from various cities in Indonesia that were close to the Bebras Bureau location. Bebras Indonesia NBO (National Board Organization) is working with volunteers who are faculty members from universities that offer informatics degrees that are willing to volunteer to train teachers in introducing Bebras.

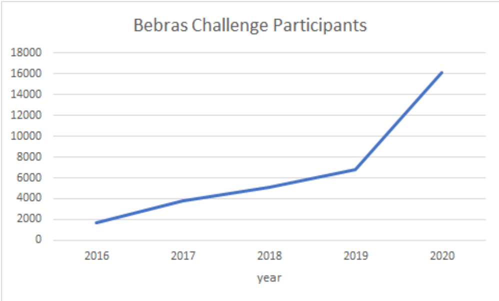


Figure 1. Bebras Challenge participants

A group of volunteers from a university gathered in the Bebras Bureau. In the year 2016, there were 12 bureaus. Nowadays, at the beginning of 2021, there are 86 Bebras Bureaus. Figure 2 depicts the distribution map of the Bebras Challenge participants.

Figure 2. Bebras Challenge participants distribution map

Compared to other countries with much smaller populations than Indonesia, Indonesian students' participation rate is very low. The roles of teachers are essential for increasing the number of participants. The Bebras Challenge made a breakthrough to introduce CT to Indonesian teachers. Teachers' communities started to wonder: "How can I introduce CT to students?". Teachers witnessed how the students were delighted and enjoyed the Bebras tasks since it is funny, joyful, and not compulsory (Dagiene, 2008; Vaníček, 2014).

Seeing the importance of CT as a 21-st century literacy and referring to the latest advances of informatics curricula globally (Shute,2017), in 2018, the Center of Curriculum of the Indonesian MOE formed a task force for defining the Indonesia K-12 informatics curriculum. As a first step, informatics is optional; it is given starting from the Junior High School students. However, CT is planned to be infused with other subjects for elementary school. The Indonesian informatics K-12 curriculum was released in December 2020, where Computational Thinking is its foundation. The curriculum integrates computational thinking, technology, and core informatics concepts (hardware, network, data analysis, algorithm and programming, and social impact of informatics). The curriculum also pays attention to Core Practices. More teachers demand CT and informatics training, which increases the growth of Bureaux significantly, including lecturers from the faculty of education. Together with the NBO, the bureau becomes a solid network of volunteers to introduce CT for Indonesia. This school-university collaboration is mutually beneficial. Teachers are facilitated by experts (information, educators) of the domain from higher education in their neighborhood. The university carries out community service, one of the obligations of universities in Indonesia to get accreditation. Though the lecturers in universities have a heavy teaching load due to COVID, mentoring the teachers in the CT training is challenging because the teachers are expressing their willingness to learn and their gratitude. The fact that informatics is planned by the Ministry of Education as a mandatory subject for Junior Highschool and the first year of high school shortly has also fueled their enthusiasm. More than 400 lecturers have joined the PANDAI movement as volunteers

4. GOOGLE SUPPORT

In 2019, Google Indonesia supported Bebras Indonesia to run a pilot project for training 150 teachers in Yogyakarta and Bandung regions to prepare teachers to implement the Indonesian informatics K-12

Google.org, the Google charity organization, granted Bebras Indonesia the funding to train 22,000 elementary and junior high school teachers during 2020 and 2021. The program is called the "PANDAI" movement, "Teacher from Digital era" in Bahasa Indonesia. The training includes an introduction to CT through the Bebras challenge, High Order Thinking Skills development, and the development of HOTS tasks related to the teachers' subject. For those who are talented and interested, programming training is held after the basic training. Considering a very large target of teachers, the training is divided into packages. A package is a project unit that consists of 40 teams. Each team consists of 11 teachers, led by a team leader. The training in one package is started with a Training for Trainers for 40 team leaders. Each team leader will create a small teacher community in his/her school that will learn CT together and infuse CT in their lectures. Teachers teach CT to the students using Bebras challenge and CT- based learning activity. However, some universities manage to train all teachers by online sessions, taking the benefit of distance learning.

One or two university faculty members are mentoring one team. With this hierarchical organization, the national-level management is more manageable. There are now 16 Packages currently running, and 24 packages are under preparation and ready to start in May 2021.

When we signed the contract, we planned to delivers our training as face-to-face classes. Due to the pandemic, the PANDAI movement has to switch its strategy to online distance learning. Though teachers' load is higher than normal due to pandemic and distance learning, they are accustomed to the online environment. Therefore, many teachers are interested in our webinars and joining the program. To introduce CT (thinking skill) not by face to face is more difficult and challenging, especially for teachers in remote areas in Indonesia. We still hope to deliver blended learning in the following months or even face-to-face training for some regions that have limited internet connections.

The Special Region of Yogyakarta is one of the provinces in Indonesia that has good achievements in education. Yogyakarta has an excellent educational culture and is well known as a city of education. In general, it has teachers with open minds and performance above the national average (Kemendikbud,2016). These characteristics make it easier for the Bebras Bureau to disseminate the importance of Computational Thinking to teachers.

Deployment of the CT model in schools is dependent on the teacher because the teacher is at the forefront of dealing with students to provide learning about CT. The Bebras Bureau invited teachers to join in this PANDAI movement through the Subject Teacher Community (in Indonesia, called MGMP). MGMP of Informatics was very enthusiastic about joining the PANDAI movement. These teachers believe that CT has become a basic skill for students in the future, and sharpening computational thinking on students will enable them to solve problems better.

Part of the schools in Indonesia is Islamic-based schools managed by the Ministry of Religion. Local authorities of the Ministry of Education and Ministry of Religion are also very supportive by providing letters of encouragement to the school under their coordination to participate in this activity. The Ministry of Religion Regional Office also appointed 480 of the best teachers to participate in this activity, not only Informatics teachers but also non-informatics teachers like science, math, and social science. Currently, around 2500 teachers are involved in the PANDAI movement in Yogyakarta, which is expected to provide this knowledge to their students. As a projection, with one teacher teaching approximately 100 students, 25,000 students will receive CT learning.

On the island of Lombok, West Nusa Tenggara province, teachers are also enthusiastic about joining the PANDAI movement. This area is a green zone in this Covid pandemic so that the implementation of workshops and training could be carried out offline face-to-face meetings. The workshop was held at the city education office. Currently, around 300 teachers are participating in this movement which is expected to give CT lessons to 3000 students.

Until mid-April 2021, the PANDAI movement has conducted more than 100 Computational Thinking workshops attended by more than 18300 teachers. On Youtube, our introduction to CT webinars to teachers has more than 176K views. Last year's Bebras Challenge by 16186 students. As a projection, by the end of 2021, it is expected that there will be more than 25,000 teachers who have attended CT workshops and more than 2 million students exposed to CT. Detailed project statistics can be seen at http://pandai.bebras.or.id.

5. CONCLUSION

It has been proven that Bebras CT Challenge has been a trigger for introducing CT to formal education. Bebras challenge creates learning motivation and a pleasant learning atmosphere for teachers and students, so it is well suited for introducing informatics naturally

Teacher motivation is fundamental but should be organized to a teacher's forum. The CT deployment in Indonesia was started from a small population of informatics teachers, growing to non-informatics subjects. CT Training for non-informatics teachers needs special effort.

The role of universities (informatics and education) is essential for preparing teachers for curriculum reform, such as what happened in Indonesia.

Lesson learned for reaching a mass population of teachers: starting small and growing with the support of other organizations such as Google. Communication and collaboration are two of the 21-st century skills that we want our students to master. By the PANDAI movement, teachers are the role model for practicing those skills. By having a more and more bureau, Bebras NBO as the national organizer has many extensions in many regions/cities in Indonesia so that the traveling cost can be minimized and become scalable to a national movement.

Acknowledgment. The three authors of this paper, who manage the PANDAI movement, express our gratitude to Bebras International, especially its Founder, Prof Valentina Dagiene, and Google.org, for their support.

6. REFERENCES

Bebras. (n.d.). *Bebras, International Challenge on Informatics and Computational Thinking*. Retrieved January 3, 2021, from https://bebras.org

Dagienė V., Futschek G. (2008) Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In: Mittermeier R.T., Sysło M.M. (eds) Informatics Education - Supporting Computational Thinking. ISSEP 2008. Lecture Notes in Computer Science, vol 5090. Springer, Berlin, Heidelberg. https://doi.org/ 10.1007/978-3-540-69924-8_2.

International Olympiad in Informatics. https://ioiinformatics.org/

Kemendikbud, (2016) Analisis Sumber Daya Manusia Pendidikan Dasar dan Menengah 2015/2016. Jakarta: Pusat Data dan Statistik Pendidikan dan Kebudayaan.

OECD. (2018). What 15-year-old students in Indonesia know and can do. In Programme for International Student Assessment (PISA) Result from PISA 2018. https://www.oecd.org/pisa/publications/PISA2018_CN_IDN.pdf

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational Research Review, 22, 142–158. doi:10.1016/j.edurev.2017.09.003

Vaníček J. (2014) Bebras Informatics Contest: Criteria for Good Tasks Revised. In: Gülbahar Y., Karataş E. (eds) Informatics in Schools. Teaching and Learning Perspectives. ISSEP 2014. Lecture Notes in Computer Science, vol 8730. Springer, Cham. https://doi.org/10.1007/978-3-319-09958-3_3

Wing J. M. (2012) Computational thinking. Microsoft Asia Faculty Summit. Tianjin, China. https://giiestemb.files.wordpress.com/2020/09/computational-thinking-jeannette-m.-wing-2012-microsoft-research-asia-faculty-summit.pdf

Computational Thinking and STEM/STEAM Education

Computational Thinking in the Mathematics Classroom

Tzi Yew Samuel LEE[1], Wen Qi Jovita TANG[2*], Hee Tee Robin PANG[3]
[1,2,3]School of Science and Technology, Singapore
lee_tzi_yew_samuel@sst.edu.sg, jovita_tang@sst.edu.sg, pang_hee_tee_robin@sst.edu.sg

ABSTRACT

Computational Thinking (CT) skills are increasingly important in the digital world. Some, such as Buitrago Flórez et al. (2017), have proposed that CT skills should be taught at the secondary school level. CT is "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (Cuny, Snyder, & Wing, 2010). According to Wing (2006), learning by computational thinking as a fundamental skill will improve the students' abstract thinking, algorithmic, and logical thinking. They will also be more all ready to solve complex and open problems. Some teachers from the School of Science and Technology, Singapore have incorporated CT in their mathematics classroom throughout the 2020 academic year. They also conducted school-based research study on whether the learning of CT was enhanced when solving mathematical problems with coding, and whether the learning of mathematical concepts can be enhanced when CT is infused. This paper focuses on some of their findings.

KEYWORDS

Computational Thinking, Mathematics, Coding, STEM

1. INTRODUCTION

Over the past few decades, technology has radically transformed the modern world. People are more dependent on computer technology. As a result, the workforce needs to have a firm grasp on computational thinking, which is "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" (Cuny, Snyder, & Wing, 2010).

School of Science and Technology, Singapore (SST) is a Specialised Independent School which offers a distinctive 4-year GCE O-Level programme with an integrated approach to applied learning. Students will be taught Python programming during the first 4 weeks of Secondary 2, which will add up to a total of 400 minutes (6 hours 40 minutes) of training. This is just enough to equip them with the fundamental programming. However, due to the limited lesson time, the emphasis is mainly skills-based and there is relatively little emphasis on the four key concepts of computational thinking -- Decomposition, Pattern Recognition, Abstraction and Algorithm.

According to Wing (2006), learning by computational thinking as a fundamental skill will improve the students' abstract thinking, algorithmic, and logical thinking. They will also be more all ready to solve complex and open problems. By incorporating CT in mathematics lessons, it is hoped that students not only have a better understanding of CT, they also deepen their understanding of mathematical concepts and processes . This is also supported by other researchers such as D. Weintrop et al. (2016), who propose that Computational thinking and mathematics have a reciprocal relationship, computation used to enrich mathematics and science learning, and applied mathematics and science contexts used to enhance computational learning. This paper focuses on some of the findings by the teachers as they explored incorporating CT in their Mathematics classrooms.

2. PURPOSE OF STUDY

The research questions are as follows:

1. Can mathematical tasks be redesigned to enhance computational thinking?
2. Can computational thinking help students to deepen their mathematical concepts and processes?

3. METHODOLOGY

A descriptive qualitative research was conducted. 51 Secondary 2 students of mixed to high ability in a Specialised Independent Secondary School participated in the study. Students were given a mathematical problem on quadratic functions to solve within two consecutive math lessons (with a total of 2 hours). Students had already been taught the basic concepts of quadratic functions, and so the lesson objective was to synthesize all they had learnt and to apply it to a problem. The students had to write a program in the Python language. The problem was subdivided into 6 tasks, which were arranged in order of their complexity. They could work on the problem individually, or in groups of no more than three people of their choice. The worksheet had a series of questions that addressed the four components of computational thinking – decomposition, pattern recognition, abstraction and algorithmic design. These questions were meant as a scaffold to guide students solve the problems by making their computational thinking processes explicit. At the end of the 2 lessons, the students had to submit the worksheet, their Python program (comprising the 6 sub-tasks) and also an individual online survey and reflection.

In this lesson study, there were a total of 21 groups (labelled GRP01 to GRP21) that were formed by the 51 students. Data from 21 Python programs, 21 sets of worksheets and 51 individual student reflections were analysed.

A second study with 59 Secondary 3 Computing students was conducted. These students have been taught about CT as part of the GCE O Level Computing syllabus. 34 groups (GRP22 to GRP55) comprising one to three members were being formed.

4. DESCRIPTION OF MATH TASK

The problem is as follows:

Quadratic functions in the general form is as follows,

$$y = ax^2 + bx + c$$

Write a computer program that requires the user to key in the values of a, b and c, where a, b and c are real numbers. The computer should output as much information as possible about the graph.

The suggested way to decompose the problem to different subtasks is as follows: [Task 1] whether the graph is concave upwards or concave downwards; [Task 2] whether the turning point is minimum or maximum; [Task 3] the y-intercept; [Task 4] the x-intercept (if there are no x-intercepts, the computer must indicate so); [Task 5] the line of symmetry; and [Task 6] the coordinates of the turning point.

In the mathematical point of view, Tasks 1, 2 and 3 are the easiest to determine. Task 4 is harder, but most students are still familiar with the mathematical formula determining the solutions. What is unfamiliar to students will be the output of two possible solutions. For Tasks 5 and 6, they involve concepts that are harder to understand.

5. FINDINGS

5.1. *Findings from Python Programs*

The 21 Python programs from the Secondary 2 groups were marked and analysed. Out of the 21 groups, 3 groups could not submit programs that were functioning properly and thus were not graded. Out of the 21 groups, 17 (81.0%) groups could complete Task 1. 16 groups (76.2%) could complete Task 2 and Task 3. 14 (66.7%) groups could complete Task 4. For Task 5 and Task 6, 6 (28.6%) groups could complete it. For groups that got Task 4, Task 5 and Task 6 wrong, the majority could still have their programs working. That is indicative that their syntax was correct, though their formula used was incorrect.

Of the 34 Python programs from the Secondary 3 groups, 26 (78.2%) completed Task 1 and Task 3, 28 (80.0%) completed Task 2, 21 (63.6%) completed Task 4, 24 (54.5%) completed Task 5 and 22 (50.9%) completed Task 6. While we see a similar general trend as the Secondary 2s in terms of the relative difficulty of each task, the completion rate for Tasks 5 and 6 were better for the Secondary Threes. This may be indicative of more familiarity and stronger internalisation of the use of pattern recognition to find the line of symmetry and turning point.

What is interesting to note is that 6 Secondary Two groups and 9 Secondary Three groups, a total of 15 (27.3%) groups also considered the case where $a = 0$, even though this was not mentioned in the question. This is a special case, as if students do not consider this option, a runtime (dividing by 0) error will occur. This indicated that students started to synthesize knowledge on their own through the CT technique of decomposition.

Clearly, the tasks were manageable to most students, with Task 5 and Task 6 being more challenging and differentiating. Nevertheless through the programs, students demonstrated they could analyse and synthesise the questions, resulting in correct solutions for the problem.

5.2. *Findings from Math Worksheets*

The worksheet had questions that addressed the four components of computational thinking – decomposition, pattern recognition, abstraction and algorithmic design. Through the answers that the students gave, it is evident that computational thinking has taken place as a whole. The sentences quoted from students below have been edited grammatically for greater clarity.

For decomposition, the worksheet has already done a great deal of scaffolding with the 6 tasks already explicitly stated to the students. However, some groups were able to make further observations on the difficulty levels between the different sub-tasks. A group noticed that

the coding comprises simple steps and more complex steps. For example, it is simple to determine the graph as concave upwards or downwards, but other steps such as finding the x-intercepts go further into the involvement of math equations. (GRP12)

Another group reported,

It is easy to generate the equation "$y = ax^2 + bx + c$". It is easy to generate the concave. It is easy to generate the y-intercept. It is difficult to generate the x-intercept(s). It is difficult to generate the line of symmetry. It is difficult to generate the turning point. (GRP18)

As earlier noted, 27.3% of the groups considered the case where $a = 0$ in their program. While considering how to decompose the problem, one group decided to

split the code into three main body parts, one for when the Discriminant is 0, >0 or <0. (GRP49)

In doing this, they noted that each case has some different properties compared to the others and may require different handling.

All three groups were clearly considering the decomposition of the main problem, with analysis on the subcomponents. On the other hand, when asked about how breaking down the problem could be useful in solving it, several groups answered generically. For example, a group answered,

Breaking down a complex problem or system into smaller parts that are more manageable and easier to understand. The smaller parts can then be examined and solved, or designed individually, as they are simpler to work with. (GRP14)

For pattern recognition, a group noticed, *the values for a, b and c can be used in the quadratic formula, completing the square as well as a general form. Hence, we just need to replace the values a, b and c in each of the different formulas to get our information. The value of c will be the y-intercept. Whether a is positive or negative will determine whether the graph is concave or convex. (GRP16)*

Another group observed,

There is a lot of conditionals to solve the problems, such as IF, ELIF, ELSE… There will be a lot of errors if you only take into consideration directly the question and do not think about math errors like the square root of a negative number…(GRP01).

Some groups also drew connection between the first two tasks, like this group which reported,

the first 2 questions could be answered together because if the graph is concave upwards, the turning point would be minimum, whereas a concave downward graph has a maximum point. for Task 5 about the line of symmetry and task 6 about the turning point of the graph, the turning point lies on the line of symmetry, so essentially, the x value would be the same and now I only need to find the y value for Task 6. (GRP11)

The students indeed observed some patterns and relationships in the tasks.

For abstraction, students could identify the crucial information that they need to process, like what this group observed,

I think the most important information is the value of a because if it is positive, the graph would have a minimum point, while if it is negative, all the other information would be different such as the graph having a maximum point.(GRP03)

Many groups also saw how one task could lead to another as well as find patterns which were common across tasks:

Taking the two roots of a quadratic graph [Task 4] and dividing it by 2 will get us the line of symmetry [Task 5]. The x value of the maximum or minimum point [Task 6] is also the value for line of symmetry [Task 5]. (GRP34)

Some groups also managed to see the limitations of computers, at least with respect to their level of programming knowledge.

We can cross factorise the values given, but it is too difficult to code. We could also use the 'completing the square' form, but it would be more complicated to substitute in the values and find the different intercepts. (GRP18)

A few groups also indicated how they could simplify the problem from the programmer's point of view, like the group below.

We can record the important information using # (comment section) beside each line of code so that we will not get confused and we will be clearer of what we are doing.(GRP09)

For algorithmic design, students could come up with ways to solve the problem, even though not all steps could be correct, like this one:

First, find the y-intercept which is c.
Substitute the value of y = 0, so ax2 + bx + c = 0
Use the quadratic formula: x=(-b ±√(b^2-4ac))/2a
Solve for x to find x-intercept(s)
Find the line of symmetry x3= (x1 – x2)÷2
y-intercept of turning point: substitute value of x into the function
Turning point: (x, y)
If a < 0, then the graph is concave downwards, otherwise it is concave upwards. (GRP18)

Some groups also could explain some of the sequences that they needed to follow to obtain the solution:

As for the other 3 tasks (Task 4, 5 and 6), they somewhat work together to give the final solutions for all 3 tasks. for example, getting the x-intercepts in turn helps to find the line of symmetry as the line of symmetry is simply the average of the x-intercepts.(GRP11)

It is interesting to note that although some groups may not be strong with mathematical concepts or in programming, they can still articulate some strategies while designing their algorithm, like this:

Ask the user for values, take the equation $ax^2 + bx + c$ and complete the square. Factorise the equation $ax^2 + bx + c$. Use the values in there that represents a graph (GRP20)

Clearly, while the Python tasks demonstrated the students' ability to understand and apply the math concepts, the CT questions in the worksheet highlighted that students have applied CT to fulfil the tasks.

5.3. *Findings from Individual Survey and Reflections*

The students were given individual survey and reflection forms towards the end of the lesson. Four statements were given to students, where they were to grade each statement on a standard 4-point Lickert Scale (1: Strongly Disagree, 2: Disagree, 3: Agree, 4: Strongly Agree). The statements and the percentage of Secondary Two and Three students who agree or strongly agree to them are shown in Table 1.

Table 1: Findings from student survey

	Statement	% of Secondary 2 Students who Agree / Strongly Agree	% of Secondary 3 Students who Agree / Strongly Agree
1	I will like to do more Computational Thinking (CT) worksheets in future.	82.4	88.9
2	For the same content and concepts covered, I prefer doing CT worksheet over the traditional pen-and-paper worksheet.	56.9	81.5
3	I am able to understand algebra (quadratic functions) better through this worksheet.	74.5	79.6
4	I am able to understand what Computational Thinking is about through this worksheet.	88.2	98.1

There is strong student perception that they are able to understand what CT is about through the activity (88.2% in Secondary 2 and 98.1% in Secondary 3 students) and a majority (74.5% in Secondary 2 and 79.6% in Secondary 3) expressed that they could understand the algebra better through the worksheet. The results corresponded with the ability that the students have demonstrated in solving the various tasks in the Python programming. This might also explain why most students (82.4% in Secondary 2 and 88.9 in Secondary 3 students) are motivated to do more Computational Thinking worksheets in future. However, a significant less proportion of Secondary Two students (56.9%) have expressed that for the same content and concepts covered, they will prefer doing CT worksheet over the traditional pen-and-paper worksheet. This may be indicative that some students are still not comfortable with programming, and the traditional pen-and-paper worksheets are still relevant when imparting mathematical concepts.

This is backed by the finding that the percentage of Secondary 3 students who would prefer doing a CT worksheet over pen-and-paper is significantly higher than the percentage of Secondary 2 students. The Secondary 3 students who participated in this activity were studying the elective GCE 'O' Level subject Computing and as such have a predisposed preference for and greater exposure to programming activities. Thus, they were more comfortable applying CT and programming their solutions as compared to their Secondary 2 peers. The higher completion rate among Secondary 3s for the more difficult Tasks 5 and 6 may also have contributed to the general sense of satisfaction with such CT and programming activities.

6. CONCLUSION

The students were largely successful in applying CT to help them break down and analyze the properties of quadratic graphs, and come up with an algorithm to find these properties. They demonstrated this success through the completion of the programming tasks, with the majority of the groups completing the basic tasks, and a handful of the groups completing the 2 complex tasks that involve analyzing and synthesizing what they have learnt. That seems to strongly suggest that CT is able help the students enhance their learning of mathematical (algebraic) processes and synthesize their mathematical concepts. Furthermore, the students' answers towards the CT questions and survey questions also seem to strongly suggest that the mathematical problem has encouraged students to apply CT skills. The recommendation for future research is to explore the teaching of computational thinking in other subjects since CT integration indeed affords new approaches to mathematics problem-solving.

7. REFERENCES

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). *Changing a Generation's Way of Thinking:* Teaching Computational Thinking Through Programming. *Review of Educational Research, 87*(4), 834-860. doi:10.3102/0034654317710096

Cuny, J., Snyder, L., Wing, J.M. (2010). *Demystifying computational thinking for non-computer scientists.* Retrieved Dec, 28, 2020 from http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Making Maths Imaginable and Visible: Integrating STEM Education with Spatial Reasoning

Chi-Cheung CHING[1*], Ka-shing CHUI[2*], Jessica Tsz-shan SO[3*], Wing-man CHIU [4*], Mei-yin LO[5*]
[1,2,3,4,5] Fukien Secondary School Affiliated School, Hong Kong
ccching@fssas.edu.hk, kschui@fssas.edu.hk, tsso@fssas.edu.hk, wmchiu@fssas.edu.hk, mylo@fssas.edu.hk

ABSTRACT

With the prompt technological, economic and scientific developments, STEM skills are necessarily developed. Specifically, problem solving skill is one of the fundamental capabilities that is often underlined. In relation to the reasoning skills employed to solve problems, spatial reasoning skills are constantly found significant in STEM disciplines. Yet, the spatial reasoning is commonly known as an ability that is cognitively cultivated that people tend to think spatial problems are solved by mainly manipulating mental models. In fact, the ability involves the coordinated manipulation of both internal and external representations. This paper aims to indicate the importance of both representations through a Primary 4 mathematics lesson demonstration. Through a comprehensive and innovative lesson design that includes hands-on activities, both visual and verbal guidance, and inquiry-based pedagogical instruction, students' spatial sensemaking skill was found effectively strengthened.

KEYWORDS

Spatial reasoning, spatial visualisation, STEM education, problem-solving, inquiry-based learning

1. INTRODUCTION

In response to the evolving needs in the rapid technological, economic and scientific developments in the 21st century, the STEM curriculum (Science, Technology, Engineering and Mathematics) is designed to equip students' capability to meet the changes and challenges in society and all over the world. These abilities, as known as the 21st-century skills, emphasise life skills including "critical thinking, problem-solving, creativity, communication, cross-cultural understanding and collaboration" (Teo, 2019). Among all these skills, "STEM learning is usually situated in the context of problem-solving" (Priemer et al., 2019; as cited in Leung, 2020). The process of problem-solving involves understanding the problem; developing a plan; carrying out the plan; looking back and giving feedback (Pólya, 1945). In other words, it connects how the unknown is linked to the data, in order to obtain possible solutions. As suggested by Leung (2020), different STEM disciplines possess their own problem-solving processes such as inquiry-based learning, engineering design, computational thinking and mathematical modelling (p.4). Problem-solving skill, as the shared fundamental skill in all four disciplines, has played a central role in STEM education. In which, reasoning skill is closely in relation to problem-solving process as it engages the process of making sense of a situation in a logical manner and comes up with a conclusion. In particular, this paper focuses on spatial reasoning skill because of their significant impacts on improving STEM achievement and their profound influences in various aspects.

The effects of spatial reasoning on STEM learning has been broadly found significant (Bell et al., 1997; Baartmans & Sorby, 2000; Casey et al., 2013; Wai et al., 2009). Spatial reasoning links closely with STEM learning as spatial understanding and manipulation are required, as well as its application in daily life. This ability has generally been studied as a cognitive phenomenon and therefore it is easily manipulated in an internal representation. But both internal and external representations coordinate with each other and are equally important. This paper aims to illustrate a Primary 4 mathematics lesson in consist of both internal and external spatial representations that enable holistic spatial ability development.

2. THEORETICAL AND PEDAGOGICAL FRAMEWORK

Spatial reasoning has acted as an important reasoning skill in STEM disciplines. As defined by the National Research Council (2006), spatial reasoning involves the location and movement of the object and ourselves, either mentally or physically, in space. It solves problems by managing, transforming and analysing data, and understanding the relationships within and between spatial structures, and through various representations. This cognitiUpscaling Skills-Based Formative Assessment The Journey Towards a Student-Run Web Application Pilot on Computational Thinking Skills ve definition tends to emphasise the internal and cognitive processes which include the working memory, manipulation of mental representations, processing speed, and cognitive load. A large-scale of researches indicated that spatial reasoning contributes to the practices of STEM professionals (Dogan & Nersessian, 2010; Stevens & Hall, 1998). Spatial reasoning supports STEM learning, particularly when faced with spatial problems that require the manipulation of internal and external representations. Given that it is compulsory to face with routine diagram or model matching tasks in various fields of STEM, such as chemistry and mathematics. For instance, in process of identifying or matching molecular models, or configurating diagrams, internal representation such as mental rotation (Shepard & Metzler, 1971) and external representation such as sketches, models and gestures (Stevens & Hall, 1998; Stevens, 1999; Dogan & Nersessian, 2010) are adopted to solve those complex and novel spatial problems. Spatial reasoning also plays a central role in everyday thinking and learning such as shopping in the supermarket, cooking, packing, playing, talking and working (Hutchins, 1995; Kirsh, 1995; Scribner 1984; Wagner, 1978). These daily activities engage the use of space and the spatial arrangement of objects and representation in the environment. "People, natural objects, human-made objects and human-made structures exist somewhere in space, and the interactions of people and things must be understood in terms of locations, distances, directions, shapes and patterns." (National Research Council, 2006, p.5). It helps solve problems that bore directly with daily life. Therefore, spatial intelligence has adaptive importance and it is fundamental to build up spatial ability.

In particular, mathematics education acts as the foundation of understanding spatial concepts, tools of representation, and processes of reasoning. The interplay between mathematics learning and spatial reasoning allows students to explore the different spatial and geometrical structures and link them with the application in daily life. More importantly, the spatial ability is transferable that its application is not exclusive to only the mathematical aspect of studies. It can be widely employed in the various fields especially for STEM disciplines which often requires 21st-century skills including problem-solving skill, computational think, critical thinking, creativity, communication and collaboration skills. According to the Hong Kong Education Bureau (2016), STEM education emphasized "nurturing students' creativity, collaboration, problem-solving skill and to foster their innovation and entrepreneurial spirit as required in the 21st century". It is believed that spatial reasoning skill is indispensable in today's STEM education.

However, spatial education and its instructions are often found "de-emphasised spatial reasoning in favor of verbal or analytic approaches to knowledge" (Ferguson, 1992; Ramey et al., 2020). It could potentially because of the cognitive nature of spatial thinking skill that makes the area of studies relatively abstract and metaphysical. In other words, this perception is made because the internal representations are often underlined rather than the external representations. But in fact, both representations are equally important.

Spatial reasoning includes four main types of skills which are intrinsic-static, intrinsic-dynamic, extrinsic-static and extrinsic-dynamic skills (as shown in Figure 1). According to Hodgkiss et al., (2018), intrinsic-static skills refer to the processing of objects or shapes, or parts of objects or shapes, without further transformation, whereas intrinsic-dynamic skills involve the processing and manipulation or transformation of objects or shapes, such as mental rotation. On the other hand, extrinsic-static skills involve understanding abstract spatial principles, processing and encoding of the spatial relations between objects. And extrinsic-dynamic skills "involve visualizing an environment in its entirety from a different position" (Shepard & Metzler, 1971).

In the lesson, the well-known mechanical cube-based puzzle by Piet Hein, SOMA, is used as the major model of instruction. The teaching design has taken an inquiry-based approach which involves not only the intrinsic-static and intrinsic-dynamic skills such as dis-embedding and mental rotation but also the extrinsic-static and extrinsic-dynamic skills. Through the hands-on activities, students are allowed to spatialise and visualise the contents and consequently develop their spatial ability.

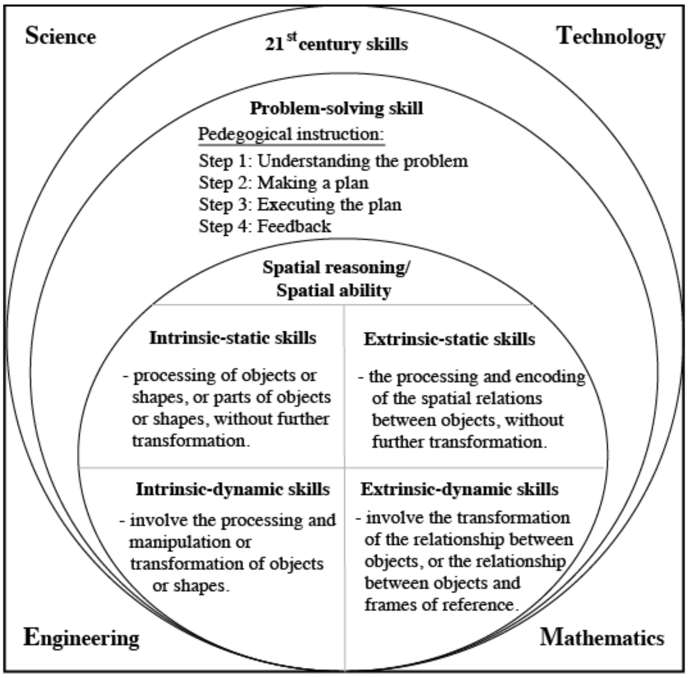


Figure 1. Theoretical and pedagogical framework of STEM education in relation to the spatial ability

3. LESSON DESIGN

The lesson adopted the model SOMA, which is a cube-based geometric puzzle designed by Piet Hein in the 1930s. The lesson could be conducted with the main activity and an extended activity along with the instruction given follows the problem-solving processes as proposed by Pólya (1945):

(A) Activity 1: Explore the 7 different types of modules

Step 1: Understanding the Problem
At the beginning of the lesson, the teacher goes through the requirements of making those modules.

The conditions are as follows:

- Each module contains no more than four cubes.
- Each module is different. (If a module could be rotated to look the same as another, it does not count as different.)
- The cubes must join a full square face when they make modules. There is no partial overlap of squares.
- The module should not be a rectangular prism.

To further help students understand the problem, the teacher delivers a preparation lesson, which allows students to explore possible rectangular prisms by assembling the cubes. The lesson preparation worksheet also provides guidance for making a prism with 1-6 cubes. Students give the answer by placing the model that they make onto the worksheet. Diagrams and photos are introduced to the students when the teacher explains the conditions to stimulate their curiosity and learning interest. The preparation lesson allows students to engage in a hands-on experience of making cubes models. Particularly with the condition of "the module should not be a rectangular prism", students could realise what the activity is leading them to, which at the end they are expected to assemble a cubic model by using those components they make at this activity.

Step 2: Making a Plan

Verbal guidance and a self-checklist are given to students to direct students' thinking. Teacher demonstrates the first two questions in the lesson activity worksheet so that students get to think about whether the module is possible to be built by 1 or 2 cubes. In this process, students adopt intrinsic-static skills to identify spatial features of the cubes and also intrinsic-dynamic skills to piece the cubes together into more complex configurations. They could think and explain if the conditions are being violated or not. Teacher makes good use of a larger size of cube model to illustrate a few possible combinations of the modules to help students visualise the modules.

Step 3: Executing the Plan

Students work in groups to find out possible modules which are built by 3 to 4 cubes with the use of the checklist. They give the answer by placing the possible modules on the worksheet. Students are required to use extrinsic-static skills to understand the spatial conditions given and compare the module structures to make sure there are no duplications.

Step 4: Giving Feedback

Once a group has completed the task, teacher display the 7 possible modules found by the group to the whole class by using a live broadcast device. So that teacher and students could assess their work together and give them instant feedback. The process of confirming if the modules are assembled correctly enables students to identify the spatial arrangement of the modules and structures. Students determine if the modules meet the conditions by using the checklist and use it as the justification.

(B) Extended Activity: Use the 7 discovered modules to build a cube and record the step of the solutions

Step 1: Understanding the Problem

The extended activity worksheet has shown the 7 modules with a labelled number to each of the specific modules (as shown in Figure 2). The numbers help to identify the combination students used later. Students are asked to use all these 7 modules to assemble a 3x3x3 cube and they are told there are many ways to form the cube.

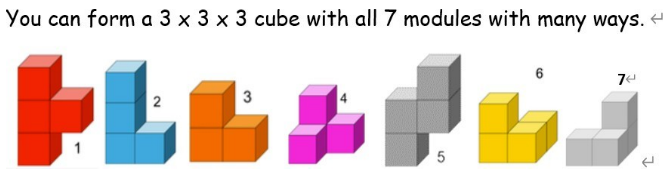


Figure 2. The 7 modules of SOMA

Step 2: Making a Plan

The extended activity worksheet guides students to record the process of combination (as shown in Figure 3). This record allows students to review how they are making sense of the combination.

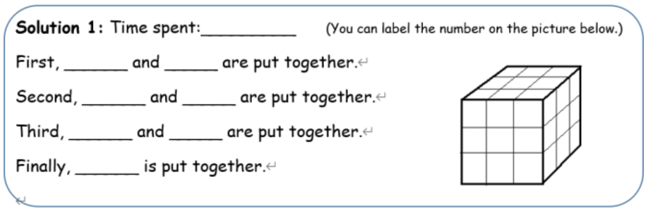


Figure 3. Guided procedures on extended activity worksheet

Step 3: Executing the Plan

The student starts assembling the modules into a cube and records their trials on the worksheet. As relocation of the modules is involved this time, the task requires students to think from a different perspective. This process engages the use of extrinsic-dynamic skills to transform the modules into another structure of the cube. It involves a coordinated manipulation of the internal representation (i.e. mental rotation) and external representation (i.e. the modules).

Step 4: Giving Feedback/ Giving Further Challenge

As mentioned earlier, there is more than one combination to form a cube. Students are further asked to give another solution once they have provided their first solution. There was one group that could find their second solution and they were asked to present their approach to form another cube.

4. OUTCOMES AND DISCUSSION

All groups were able to unlock the modules successfully in the first activity and moved on to the extended activity. Two outstanding groups were able to finish the extended task in a short period of time. One group of students were able to find two solutions and presented their approach in class. More importantly, students enjoyed the lesson a lot and share lots of insightful thoughts and ideas throughout the discussion section.

4.1 Pre-test and Post-test Analysis on Students' Spatial Ability

A pre-test and post-test were conducted to evaluate students' spatial reasoning skills by identifying if the two models given are identical (as the example shown in Figure 4). Four questions are designed to check their ability to recognise the features of the models (i.e. intrinsic-static skills), to differentiate the two models (i.e. intrinsic-dynamic skills), to identify the missing component of the model compared to the other model (i.e. extrinsic-static skills), and to tell how do different sides of the models look like (i.e. extrinsic-dynamic skills).

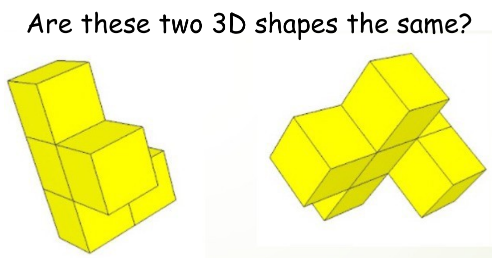


Figure 4. An example question of the pre-test and post-test

The mean scores are 3.07 and 3.52 out of 4 marks for the pre-test and post-test respectively (Table 1). It implies that there is a significant improvement after the pedagogical instruction was input. It also indicated that the spatial reasoning skill could be systematically trained by imagining and visualizing the spatial models. In other words, their spatial sensemaking is contributed by both internal and external spatial training.

Table 1. Pre-test and post-test results on students' spatial ability

	Mean	S.D.
Pre-test (N=24)	3.07	1.32
Post-test (N=24)	3.52	0.71

4.2 Students' Qualitative Feedback on the SOMA Lesson

Apart from the quantitative feedback received from students, their comments on the SOMA lesson are also reflective:

"I think the SOMA lesson is very interesting because I can learn different ways to turn over the shape and discover how blocks can create a 3-D shape. I can also improve my non-verbal reasoning skills. It is fun to learn things that are not from the Maths textbook. I would like to have another SOMA lesson."

"I like the SOMA lesson because there are cubes that can make me imagine it more easily."

"I like the SOMA lesson because, through this lesson, I've learnt more able 3D shapes than before. And I've also learnt how to transform different shapes. I hope that the school can have more SOMA lessons."

"I like SOMA because it let us experiment with how to make the cubes."

"I like the SOMA lesson very much because I can think more about how to combine it and the correct shape. It is so fun. I think this is the best mathematics lesson."

These comments reflect that students are well aware of the lesson objects as some of them could tell the skills they have adopted to complete the tasks. Additionally, students are satisfied with the lesson because most of them found the tasks challenging but at the same time inspiring and innovative.

5. CONCLUSION

The presented lesson focus on spatial ability which supports solving spatial problems. This lesson emphasises the importance of using both internal and external spatial representations. Though both visual and verbal guidance, students' spatial sensemaking skill is effectively strengthened.

To further extend students' learning, a person-height large SOMA model is planned to build and display in the school. It is also planned to establish a spatial museum where students' works will be exhibited. The models are not just limited to SOMA as other figures will be gradually infused into different levels of curriculum as well. It is believed that making learning visible along with the internal processing essentially promote comprehensive spatial development.

6. REFERENCES

Baartmans, B., & Sorby, S. (2000). The development and assessment of a course for enhancing the 3-D spatial visualization skills of first-year engineering students. Journal of Engineering Education, 89, 301–307.

Bell, J. E., Hsi, S. & Linn, M. C. (1997). The role of spatial reasoning in engineering and the design of spatial instruction. Journal of Engineering Education, 86, 151–158.

Byrne, R. M., & Johnson-Laird, P. N. (1989). Spatial reasoning. *Journal of memory and language*, *28*(5), 564-575.

Casey, B., Dulaney, A., Sorby, S., & Veurink, N. (2013). The role of spatial training in improving spatial and calculus performance in engineering students. Learning and Individual Differences, 26, 20–29.

Dogan, F., & Nersessian, N. J. (2010). Generic abstraction in design creativity: The case of Staatsgalerie by James Stirling. Design Studies, 31, 207–236. http://dx.doi.org/10.1016/j.destud.2009.12.004

Ferguson, E. S. (1992). Engineering and the Mind's Eye. MIT press.

Hart, G. and Heathfield, E. (2017). *Making Math Visible*. [online] Makingmathvisible.com. Available at: <http://makingmathvisible.com> [Accessed 2 January 2021].

Hodgkiss, A., Gilligan, K. A., Tolmie, A. K., Thomas, M., & Farran, E. K. (2018). Spatial cognition and science achievement: The contribution of intrinsic and extrinsic spatial skills from 7 to 11 years. *The British journal of educational psychology*, *88*(4), 675–697. https://doi.org/10.1111/bjep.12211

Höffler, T. N. (2010). Spatial ability: Its influence on learning with visualizations—a meta-analytic review. *Educational psychology review*, *22*(3), 245-269.

Hong Kong Education Bureau. (2016). *Report on Promotion of STEM Education Unleashing Potentialin Innovation.* Retrieved 1 November 2019, from https://www.edb.gov.hk/attachment/en/curriculum-development/renewal/STEM%20Education%20Report_Executive%20Summary_Eng.pdf

Hutchins, E. L. (1995). Cognition in the wild. Cambridge, MA: MIT Press.

Kirsh, D. (1995). The intelligent use of space. Artificial Intelligence, 73, 31–68.

Leung, A. (2020). Boundary crossing pedagogy in STEM education. *International Journal of STEM Education*, *7*, 1-11.

Lowrie, T., Logan, T., & Hegarty, M. (2019). The influence of spatial visualization training on students' spatial reasoning and mathematics performance. *Journal of Cognition and Development*, *20*(5), 729-751.

McLeod, D. B. (1989). The role of affect in mathematical problem solving. In *Affect and mathematical problem solving*(pp. 20-36). Springer, New York, NY.

National Research Council. (2006). *Learning to think spatially: GIS as a support system in the K-12 curriculum.* Washington, DC: National AcademicPress.

Pólya, G. (1945). *How to solve it*. Princeton: Princeton University Press.

Raje, S., Krach, M., & Kaplan, G. (2013). Connecting spatial reasoning ideas in mathematics and chemistry. *MatheMatics teacher*, *107*(3), 220-224.

Ramey, K. E., & Uttal, D. H. (2017). Making sense of space: Distributed spatial sensemaking in a middle school summer engineering camp. *Journal of the Learning Sciences*, *26*(2), 277-319.

Ramey, K. E., Stevens, R., & Uttal, D. H. (2020). In-FUSE-ing STEAM learning with spatial reasoning: Distributed spatial sensemaking in school-based making activities. *Journal of Educational Psychology*, *112*(3), 466.

Scribner, S. (1984). Studying working intelligence. In B. Rogoff & J. Lave (Eds.), Everyday cognition: Its development in social context (pp. 9–40). Cambridge, MA: Harvard University Press.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. Science, 171, 701–703.

Shumway, J. F. (2013). Building bridges to spatial reasoning. *Teaching Children Mathematics*, *20*(1), 44-51.

Stevens, R. R. (1999). Disciplined perception: Comparing the development of embodied mathematical practices in school and at work (Doctoral dissertation). University of California, Berkeley, CA.

Stevens, R., & Hall, R. (1998). Disciplined perception: Learning to see in technoscience. In M. Lampert & M. L. Blunk (Eds.), Talking mathematics in school: Studies of teaching and learning (pp. 107–150). Cambridge, UK: Cambridge University Press.

Teo, P. (2019). Teaching for the 21st century: A case for dialogic pedagogy. *Learning, Culture and Social Interaction*, *21*, 170-178.

Thorndyke, P. W., & Goldin, S. E. (1983). Spatial learning and reasoning skill. In *Spatial orientation* (pp. 195- 217). Springer, Boston, MA.

Wagner, D. A. (1978). Memories of Morocco: The influence of age, schooling and environment on memory. Cognitive Psychology, 10, 1–28.

Wai, J., Lubinski, D., & Benbow, C. P. (2009). Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of educational Psychology*, *101*(4), 817.

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Computational Thinking in Mathematics (Grade 2-6): Developing CT Skills and 21st Century Competencies

Felicia CHOON[1], Staphni SIM[2]*
[1,2]HORIZON Primary School, Singapore
Choon_yoke_chan_felicia@schools.gov.sg, sim_yan_ling_staphni@schools.gov.sg

ABSTRACT
The Computational Thinking in Mathematics (CTIM) program at Horizon Primary School (HRPS) has been enabling students from grade 2 through 6 to develop computational thinking skills through age-appropriate computer coding lessons and projects. Students have become more confident and show greater resilience in persisting to solve Mathematical and task-based problems.

KEYWORDS
Computational thinking, Coding, Problem solving, Mathematics, 21st century competencies

1. INTRODUCTION
Computational thinking (CT) is a way of approaching situations in a systematic manner for effective and efficient problems solving. As Wing (2006) stated, "Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation."

Beyond CT skills, we want students to acquire collaborative skills, and dispositions such as confidence and resilience. These are important 21st Century competencies that we want students to develop during their elementary school years.

At HRPS, all grade 2 through 6 students participate in ten hours of Computational Thinking in Mathematics (CTIM) lessons yearly. By using block-based programming, CTIM lessons are designed to make use of Mathematics content-related projects to allow students to learn both CT skills and 21st Century competencies.

The presenters will share how CTIM program is carried out, the challenges that the school faced, the school's innovative practices and the observable outcomes of the program.

2. CTIM PROGRAM

2.1. *CTIM Framework*
The framework that aids in the development of CT pedagogy and assessment is derived from literature reviews and consultation with STEM Inc (a unit in Science Centre Singapore dedicated to promoting STEM education in Singapore schools). CTIM program revolves around five core CT components – Problem decomposition, pattern recognition, abstraction, algorithm design, and reflection and refinement – derived from Wing's (2006) CT cognitive processes. The 'heart' of the CTIM program is to enable students to become critical thinkers who are effective and efficient problem-solvers (Shute, 2017).

We believe that there are three key factors to successful implementation of the CTIM program. They are (1) a suitable curriculum that is customized to our school's context, (2) the attitudes of stakeholders - school management, teachers, students and parents, and (3) the building of capacity, in terms of both teachers' ability to conduct CTIM as well as the physical resources e.g. equipment to support the implementation of CTIM.

The outermost ring of the framework is the 21st Century competencies that the school wants to develop in students through CTIM.

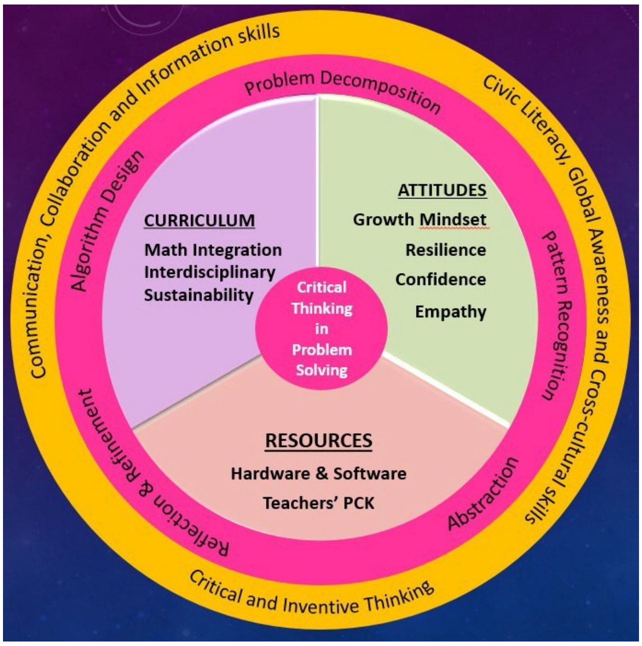


Figure 1 CTIM Framework

2.2. *CTIM Program*
The CTIM curriculum integrates and aligns coding and protoyping activities to the Mathematics curriculum to help students reinforce Mathematical thinking skills through CT processes. At the end of each year's program, students would have learned to code a microprocessor called Micro:bit using a web-based software, Makecode.org. Purposeful and age-appropriate opportunities are provided for real-life application of learning. This allows students to make meaningful connections across various disciplines.

Table 1 Summary of CTIM Program

Grade	Area of Focus	Math Component
2	Introduction to Micro:bit and coding to display numbers	Math concepts: Shapes and

	and shapes on LCD. Project: Design a game	Patterns / Multiplication
3	Code motion sensor and LCD display as output. Project: Design a coin bank	Math concepts: Multiplication of whole numbers
4	Integrate the use of moisture sensor, buzzer and LCD display using Loop and logic Project: Design a soil moisture alert system	Math concepts: Statistics (Tables and Line graphs)
5	Foregrounding CT processes – making thinking observable Project: Coding a robot to perform tasks	Math thinking – Logical and sequential thinking
6	Consolidation of coding skills and functions of peripherals in real life problem solving opportunities. Project: Based on the theme 'Sustainability', students will work collaboratively to design a product that will promote sustainable living.	Consolidation of both Math concepts and Math thinking skills

3. CHALLENGES AND INNOVATIVE PRACTICES

The Ministry of Education provided generous funding for the school to implement CTIM as Horizon Primary School's Applied Learning Program (ALP). This enabled the school to provide students and teachers with computers and Micro:bit sets during CTIM lessons.

However, the implementation of CTIM was not without its challenges. Some challenges are anticipated and as such, steps have been taken to mitigate them. Others are challenges that emerged along the way and required us to put CT skills into practice and innovate ways to overcome them.

3.1. Teachers

3.1.1. National Curriculum and Limited Time

One challenge faced in the implementation of CTIM was navigating the tension between teachers' concerns about the completion of the National Primary School Mathematics Curriculum and the implementation of CTIM program. This affected teachers' attitude towards the program and their willingness to carry out CTIM lessons.

To address their concerns, we integrated Math content and thinking skills into the CTIM lessons. The lessons are designed such that students will deepen their understanding of Math concepts and develop their Math thinking skills by applying them to complete the tasks assigned. Integrating computing into content areas increases access to computational experiences and provides a way of introducing computing within authentic experiences rather than as isolated subject areas (Jona et. al., 2014 as cited in Israel et. al., 2015).

3.1.2. Computing Competencies and Confidence

Teachers were apprehensive about teaching CT skills for coding Micro:bit as they had not received any formal training on how to code, much less how to teach. Without relevant pedagogical content knowledge (PCK) and the technical expertise of computing, teachers expressed a lack of confidence in implementing CTIM program initially.

To build teachers' competencies to teach CT skills through computing, we provided three layers of training.

(1) A small group of about fifteen teachers received training conducted by STEM Inc to prepare them to become CTIM trainers.

(2) CTIM trainers conducted onboarding workshop for all teachers of the school to create a baseline awareness and understanding of the CTIM program. They also taught teachers basic coding as part of the workshop experience.

(3) CTIM trainers conducted additional professional development sessions for Math and Science teachers to further enhance their competencies to conduct CTIM lessons using the curriculum developed by the Math Department.

The school further supported teachers by deploying an ICT officer with the expertise to troubleshoot hardware and connectivity issues during the conduct of CTIM lessons. This enabled teachers to focus on teaching CTIM to students.

3.2. Students

3.2.2. Age-appropriateness

A design dilemma faced was whether CT skills should only be taught in the upper grades (grade 4, 5 and 6). The fact that CT skills and 21st Century competencies takes time to develop implied that we needed to provide students with more time and opportunities to do so. Thus, starting CTIM program at grade 2, which is after their transition year into formal schooling, would have afforded us the longest runway for students' develop of CT skills.

In the design of the CTIM curriculum, content and activities are carefully curated to ensure age-appropriateness. The ALP coordinator gathers feedback from teachers conducting CTIM regularly and refines the curriculum at the end of each year's run of the program.

3.2.2. Learning Challenges

Initially, students who came from backgrounds that provided them with little experience with computing and students who have learning difficulties found it difficult to keep up with the pace of the lessons. However, students overcame such difficulties quickly as we observed that they were often quick to pick up computing skills when teachers provided sufficient support. Students also learnt to collaborate and tap on each other's strengths in problem solving. This spirit of collaboration was more evident in among upper grade students.

We were encouraged to observe that students who normally displayed difficulty in coping with the learning of the National Mathematics curriculum reported both interest and enthusiasm in CTIM lessons. The success that these students experienced in being able to complete the coding tasks improved their general confidence in themselves as problem-solvers.

4. OBSERVABLE OUTCOMES

Qualitative feedback gathered from teachers, students and parents were generally positive.

Teachers reported that students showed improvement in their abilities to do computing and became more resourceful and confident in solving problems using CT skills after experiencing CTIM program. They expressed that the professional development sessions increased their confidence in conducting CTIM lessons.

Parents expressed enthusiasm about the program, especially after going through the Parents' CTIM Workshop where they experienced first-hand how CTIM lessons are conducted.

Students' feedback were gathered using questionnaires and surveys as they are the most commonly used measure for knowledge of and/or attitudes towards CT (Shute, 2017). A survey was conducted in 2019 to evaluate the effectiveness of CTIM program. Selected results are shared in Table 2.

- Statements 1 and 2 provide an indication of students' interest in coding
- Statements 3 to 5 provide insights into students' self-directedness and independence in learning
- Statement 6 helps the school understand whether students find CT skills useful in problem solving
- Statement 7 provides an indication of students' resilience in problem-solving

Table 2 Results of 2019 CTIM Program Survey

Answers to statements	Yes			No		
Grade level (cohort size)	3 (141)	4 (181)	5 (211)	3	4	5
1. I enjoy coding.	91	90	89	9	10	11
2. I am interested to learn other types of coding.	91	85	79	9	15	21
3. I would attempt the coding assignments on my own.	82	71	70	12	29	30
4. I will do another coding project on my own.	72	68	66	28	32	34
5. I would look at other coding examples to learn from them.	79	80	82	21	20	18
6. I have learnt useful problem-solving skills.	90	86	85	10	14	15
7. I will not give up when I encounter coding problems.	91	86	86	9	14	14

5. CONCLUSION

Without a doubt, competencies like adaptability, resilience and collaboration, and critical thinking skills are much needed for students to navigate the complexities of the VUCA world. While acquiring knowledge from various disciplines remain important, developing thinking skills and 21st Century competencies are equally critical in order for students to be future-ready.

Through the CTIM program, HRPS is able to help the students to develop CT skills and 21st Century competencies. Witnessing the students' growth in these aspects has been remarkable. Not only has the student body benefited, many of the teachers too, have moved from having no experience to a place of pride in themselves as competent CTIM teachers.

Consequently, we recognize that the current CTIM program will be continuously refined as the needs of students are ever evolving. Staying true to the spirit of CTIM program, the program designers will practise the CT processes iteratively, to review and make improvements so the program can stay relevant and effective.

6. REFERENCES

Israel, M., Pearson, J., Tapia, T., Wherfel, Q., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. Computers and Education, 82, 263–279. https://doi.org/10.1016/j.compedu.2014.11.022

Shute, S. (2017). Demystifying computational thinking. Educational Research Review, 22, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Wing, J. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. doi:10.1145/1118178.1118215

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Computational Thinking in Mathematics: Calculating Riemann Sums with Graphical Calculator and beyond

Xiajuan YE
NUS High School of Math and Science, Singapore
nhsyx@nus.edu.sg

ABSTRACT

Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent (Wing, 2011). The integration of computational thinking is possible in teaching of Riemann Sums. This paper shares a redesign of a segment of curriculum to infuse computational thinking in one of the math lessons on Riemann Sums. The lesson was conducted to enrich students' learning experiences and deepen their understanding of three approximation methods in Riemann sums, namely Left Sum, Right Sum and Midpoint Sum using the programming function feature in Graphical Calculator (GC) at hand. Students were also given opportunity to generalise the three approximation methods into one.

KEYWORDS

computational thinking, Riemann sum, Left Sum, Right Sum, Midpoint Sum

1. INTRODUCTION

Computational thinking is the key to preparing Singaporeans for the digital century and becoming Future-Ready digital citizens with Singapore's vision of Smart Nation. At NUS High School of Math and Science (NUS High), all students will be required to read CS1131 Computational Thinking in Year 1 Semester. By the end of the course, students will be able understand basic programming principles and concepts such as iterations with for loops, conditionals and variables using turtle graphics in Python (Programme of Studies, NUS High). Four key elements of computational thinking are introduced, namely, algorithm design, decomposition, pattern recognition and abstraction. The designed lesson focused on algorithm design and abstraction, where algorithm design aims to develop the step-by-step instructions for solving problems and abstraction aims to identify the general principles that could be generalised the problem under discussion.

The lesson was conducted with a class of 19 students, 15 of whom take Computer Science as a major subject. The students were asked to bring their GCs (model: TI nSpire CX CAS) to the lesson beforehand.

Riemann Sums are used to approximate the definite integral $\int_a^b f(x)\,dx$ where direct integration is challenging. The sums were introduced to students one day before the lesson. Let $f(x)$ be a function defined on a closed interval $[a, b]$ which is divided into n subintervals by the point $x_0, x_1, x_2, \cdots, x_{n-1}, x_n$ where

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b.$$

For each $k \in \{1, \cdots, n\}$, we choose $x^*_k \in [x_{k-1}, x_k]$, i.e., x^*_k is a point in the interval $[x_{k-1}, x_k]$. By writing the difference $x_k - x_{k-1}$ as Δx_k, we can form the Riemann sum

$$\sum_{k=1}^{n} f\left(x^*_k\right) \Delta x_k .$$

Quite often however, we divide the interval $[a, b]$ into n subintervals with equal length of $\frac{b-a}{n}$ to make our calculation easier. The choice of x^*_k leads to three different types of Riemann sums, Left Sum, Right Sum and Midpoint Sum. The three approximation methods were introduced to students one day before the lesson. In all three methods, rectangles are used to approximate.

$$\text{Left Sum} = \frac{b-a}{n}\sum_{k=1}^{n} f(x_{k-1})$$

$$\text{Right Sum} = \frac{b-a}{n}\sum_{k=1}^{n} f(x_k)$$

$$\text{Midpoint Sum} = \frac{b-a}{n}\sum_{k=1}^{n} f\left(\frac{x_{k-1}+x_k}{\ }\right)$$

It can be easily seen that $x_k = a + k \cdot \frac{b-a}{n}$ (Krantz, Steven G. ,1991).Students learned how to calculate the Riemann sums given a few intervals the day before the designed lesson.

2. Lesson Enactment

The lesson started with a recap of the four key elements of computational thinking with a focus on algorithm design and abstraction.

2.1 Algorithm Design

Starting from Left Sum, based on inquiry-based learning, students were guided through the thinking process.

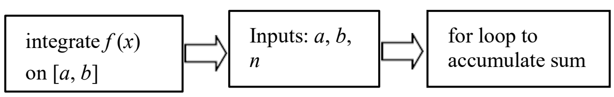


Figure 1. Guided Thinking Process

GC instructions for calculating the Left Sum were given step-by-step for students to familiarize with the simple process and syntax. Students worked on Right Sum and Mid Sum codes on their own with help from teacher just to fix some syntax errors during the lesson.

2.2 Screenshot of functions for Left Sum

```
lsumtest
Define lsumtest(a,b,n)=
Func
Local s,h,k
h:=(b-a)/n
s:=0
For k,1,n
  s:=s+f(a+(k-1)·h)
EndFor
Disp "Left Sum =",s
EndFunc
```

Figure 2. Screenshot of the function for Left Sum in GC

The screenshots of functions for Right Sum and Midpoint Sum can be found in the appendix.

2.3 Generalisation - Three Functions into One

Students were asked to discuss to use only function to calculate all three sums.

Noticing that, in the calculation of Left Sum and Right Sum, we have

$$x_{k-1} = a+(k-1)\cdot\frac{(b-a)}{n} \quad \text{and}$$

$$x_k = a + k\cdot\frac{(b-a)}{n}.$$

In the calculation of the Midpoint Sum, we have

$$\frac{x_{k-1}+x_k}{2} = \frac{\left[a+(k-1)\cdot\frac{(b-a)}{n}\right]+\left[a+k\cdot\frac{(b-a)}{n}\right]}{2}$$

$$= \frac{2a+(2k-1)\cdot\frac{(b-a)}{n}}{2}$$

$$= a+\left(k-\frac{1}{2}\right)\cdot\frac{(b-a)}{n}$$

Hints were then given that other than a, b and n, an extra input c, is needed. At first, some students identified $k - c$ in the sums above, the values that c takes are 1, $\frac{1}{2}$ and 0 in Left Sum, Right Sum and Midpoint Sum, respectively. Students observed that they are to work in the order of sum as the 'x' values increase. As a class discussion, they identified $k - 1 + c$ in the sums under discussion and c were then chosen as 0, $\frac{1}{2}$ and 1 in Left Sum, Right Sum and Midpoint Sum, respectively.

2.4 Screenshot of the generalized function in GC

The screenshot of function of the generalised Riemann sum can be found in the appendix.

3. OBSERVATION AND FINGIDNS FROM STUDENTS

Once the actual hands-on coding session started with the programming function feature, all students participated actively, with those taking Computer Science quite excited of knowing that they can code with GCs and helping those who do not take Computer Science to understand some syntax matters. For the two students who did not bring their GCs, one used Python online editor and compiler and the other used Python IDLE in his laptop.

A survey was conducted for the class of 19 students who attended the lesson and 16 of them responded.

When asked how useful students found the computational thinking lesson in helping them understand Riemann sums better and deeper on a 5-point Likert scale, 93.8% of the students chose 3 and above (with 62.5% of the students chose 4 and 5).

Here are some quotes from students when asked what they liked most about the lesson:

- *It helps us understand a bit deeper and in terms of what we know and are familiar with. It also helps us navigate the GC better.*
- *I liked thinking about the summation algorithmically and writing code to perform summation.*
- *Coding was a fun and enriching experience and applying it to mathematics.*
- *I like the innovative and new method to help us understand.*

When asked how the lesson could be improved students commented that the lesson could go further to introduce Simpson's rule. While some commented that syntax could be better summarized to facilitate the lesson and the arrange of tabs could be messy when they use the handheld GC.

4. CONCLUSION AND REFLECTION

Computational thinking allows students to be active, rather than passive, users of technology. The way we understand the technology that surrounds us, and the way we ask questions about these devices, will become a significant differentiator in the 21st-century workforce (Kristen Thorson, 2018). Out of the 19 students who attended the lesson, 16 of them are familiar with Python programming. They struggled a bit at the start when they are to code with TI-Basic (programming language in GC). Yet they were able to successfully code with it within the one and half hour lesson.

Through algorithm design, students have an opportunity to apply algorithmic thinking whenever they create or use a well-defined series of steps to achieve a desired outcome (Eli Sheldon, 2017). The algorithmic thinking also enables students to both communicate and interpret clear instructions for a reliable output.

Through the process of abstraction, students can learn to sort through all the information available to identify the specific information they need. This is an invaluable skill as students read larger texts and are presented with more and more complex information (Kristen Thorson, 2018).

The topic chosen provides a good opportunity to introduce algorithm design and abstraction in computational thinking. As all students know basic coding, the simple yet different syntax does not stand in the way of coding with GC.

The survey results show that most students gained better experience and developed some form of algorithm design and generalisation skills in computational thinking.

5. REFERENCES

Krantz, S.G. (1991), Real Analysis and Foundations.CRC Press. p. 173.

Programme of Studies (NUS High School of Math and Science). Retried on February 10, 2021, from https://www.nushigh.edu.sg/qql/slot/u744/POS_Class%20of%202021_final.pdf .

Sheldon, E. (2017). *Computational Thinking Across the Curriculum*. Retrieved on Apr 10, 2021, from https://www.edutopia.org/blog/computational-thinking- across-the-curriculum-eli-sheldon.

Thorson, K. (2018). *Early Learning Strategies for Developing Computational Thinking Skills*. Retrieved on Apr 10, 2021, from https://www.gettingsmart.com/2018/03/early-learning-strategies-for-developing-computational-thinking-skills/.

Wing, J.M. (2011), Research Notebook: *Computational thinking -what and why?* The Link Magazine, 20-23.

6. Appendix

The screenshots of functions for Right Sum, Midpoint Sum and the generalised Riemann sum can be below.

6.1 Screenshot of the function for Right Sum in GC

```
rsumtest                                  0/7
Define rsumtest(a,b,n)=
Func
Local s,h,k
h:=(b-a)/n
s:=0
For k,1,n
  s:=s+f(a+k·h)
EndFor
Disp "Right Sum =",s
EndFunc
```

6.2 Screenshot of the function for Midpoint Sum in GC

```
midsum                                    0/9
Define midsum(a,b,n)=
Func
Local s,h,k,xk1,xk
h:=(b-a)/n
s:=0
For k,1,n
  xk1:=a+(k-1)·h
  xk:=a+k·h
  s:=s+f((xk1+xk)/2)
EndFor
Disp "Midpoint Sum =",s
EndFunc
```

6.3 Screenshot of the function for Generalised Riemann Sum in GC

```
riemannsums3in1                           7/7
Define riemannsums3in1(a,b,n,c)=
Func
Local s,h,k
h:=(b-a)/n
s:=0
For k,1,n
  s:=s+f(a+(k-1+c)·h)
EndFor
Disp "Riemann Sum =",s
EndFunc
```

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Computational Thinking in Statistics

Frank NG
Republic Polytechnic, Singapore
frank_ng@rp.edu.sg

ABSTRACT

Statistics for students is usually based on the traditional methods of getting the students to learn specific techniques and memorizing statistical formulas and then applying these formulas to extract meaning. The implication is then for the students to pick up the knowledge and domain expertise in using the statistical techniques. This places a heavy burden on the students. This paper is part of a work in progress and examines the shortcomings of the traditional method of teaching a statistical technique – multiple linear regression when compared to computational thinking as a tool in teaching statistics. The computational thinking approach has the potential to enhance the generalization of statistical problems. In this qualitative paper I reflect on how computational thinking, when applied to the general linear regression model, can show the key ideas in simplifying the problem of the linear model from simple regression to multiple linear regression.

KEYWORDS

Mathematical model, generalization, machine learning, gradient descent, objective function

1. INTRODUCTION

Computational thinking (CT) (Wing, 2006) describes a set of thinking skills, habits and approaches that are integral to solving complex problems using a computer and widely applicable in the information society. Distilled down to its most fundamental elements, CT comprises four parts: decomposition, pattern recognition, abstraction, and algorithmic thinking. With these four skills, one can specify the solution to a problem, which can then be executed by a computer or a human following a set of instructions. (Looi, 2017). These key ideas were used to overcome the shortcomings of the statistical method used in the linear regression and multiple linear regression.

2. SHORTCOMINGS

Students who are first introduced to linear regression found the least square method easy to understand and apply. However, they find difficulty in applying the least square method to multiple linear regression as the extension of the statistical method to more than 2 independent variables get complicated as there is a need to calculate partial correlations for the dependent variable. Students have to learn how to work out the multiple correlation coefficients for these independent variables and as the number of independent variables increases, the calculations to find the predicted linear equation relationship between the dependent variable and the independent variables becomes very complicated and no clear pattern can be discerned to generalize a solution for any number of independent variables (Hinton, 2014).

3. ALTERNATIVE METHOD

Computational thinking in Statistics allows the students to overcome the shortcomings of the statistical approach by using an alternative method. By learning the fundamentals of linear regression and computational thinking, it allows students to generalize the results to multiple linear regression without the need for complicated calculations. The students first create a mathematical model that represents the problem and then using computational thinking and the use of MS-Excel solver, the students can derive the predicted results with the same degree of accuracy as the traditional statistical methods.

4. COMPUTATIONAL THINKING

The computational thinking concepts used here are:

4.1 Decomposition

The first step of computational thinking is decomposition – breaking down the problem into simpler parts. The problem is 'decomposed' into one independent variable X and one dependent variable Y. For the linear regression problem, this is given by the equation $Y = a + bX$. The idea is then to find an individual error e_i for each observation i which is the difference between the actual value Y_i and the predicted value $\hat{Y}_i$. This is given by $e_i = Y_i - \hat{Y}_i$ where $\hat{Y}_i = a + bX$. After looking at each individual error e_i, the squares of these errors are then summed up to obtain the objective function which is the sum of all the individual errors. This is given by $\sum E = \sum e_i^2 = \sum_{i=1}^{i=n} (Y_i - \hat{Y}_i)^2$ for all values of i.

4.2 Pattern Recognition

The students can recognize the pattern that link the input independent variables to the output dependent variable. That pattern can be easily discerned as the problem evolves from the simple linear regression $Y = a + b_1X_1$ to the multiple linear regression $Y = a + b_1X_1 + ... + b_kX_k$, as there is only the need to add in the additional independent variables to the mathematical model and then carry out the procedure as per the simple linear regression model. This approach is scalable and is very general in its approach and can find the best fit line or hyperplane for any input independent variables.

4.3 Abstraction

The students will learn how to create an abstract model that represent the linear relationship between the input independent variables $X_1,X_2,\ldots,X_k$ and the output dependent variable Y. The difference between the predicted dependent value and the actual dependent value will be the error. The general idea will be to minimize the total error by finding the coefficients of the predicted equation. The figure below shows the linear regression model.

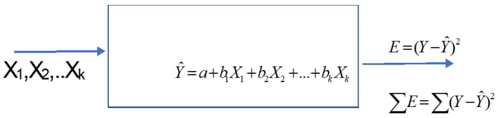


Figure 1. Linear Regression Model.

The computational approach using machine learning algorithms such as gradient descent can be used to generalized to any number of input independent variables X1,X2,..,Xk for one dependent variable Y.

4.4 Algorithmic Thinking

In order to find these values in the linear regression model, we can use a machine learning algorithm called gradient descent that will be able to minimize the objective function which is the sum of the squares of the errors (SSE) given by $\sum E$ by changing the values of a,b1,b2,..bk. Gradient descent is a well-known algorithm that will initialize the unknown values and then change these values to minimize the SSE until convergence is reached. The method relies on finding the gradients of the unknown values and then updating the unknown values by their gradients using a step function called the learning rate. We can generalize the result for more than one independent variable by including an additional independent variable. The model remains unchanged, and the equation line is now $Y=a+b_1X_1+b_2X_2$

We carry out gradient descent using Microsoft Excel Solver. Microsoft Excel Solver will compute the gradients and update the unknown values till convergence. A gradient descent, using both the traditional statistical approach and the machine learning approach, was carried out using the sample dataset of 10 students' Study Time (X) versus Exam Marks (Y)

The results of the machine learning approach agree well with the answers based on the traditional statistical approach as shown in figure 2.

a	b
34.4057	0.7446

student	Study Time (X)	Exam Marks (Y)	y=a+bx	Err Square
1	40	58	64.19093816	38.32771527
2	43	73	66.42482843	43.23288113
3	18	56	47.80907613	67.09123378
4	10	47	41.8520354	26.50153954
5	25	58	53.02148678	24.7855939
6	33	54	58.97852751	24.7857362
7	27	45	54.51074696	90.45430777
8	17	32	47.06444604	226.9375346
9	30	68	56.74463724	126.6831909
10	47	69	69.4033488	0.162690255
			ErrorSum	668.9624233

Figure 2 Linear Regression using gradient descent

This approach is scalable and is very general in it approach and can find the best fit line or hyperplane for any input independent variables. The extension of linear regression to multiple linear regression using gradient descent is shown in Figure 3.

a	b1	b2
0.3024	0.3024	0.6486

student	Intelligence Score	Study Time (X)	Exam Marks (Y)	ypredict=a+b1*intelligence score +b2*study time	Err Square
1	118	40	58	61.92546158	15.40924859
2	128	43	73	66.89490348	37.27220347
3	110	18	56	45.23688184	115.8447125
4	114	10	47	41.25731977	32.9783762
5	138	25	58	58.24324588	0.059168559
6	120	33	54	57.98980918	15.91857732
7	106	27	45	49.8650677	23.6688837
8	124	17	32	48.82125682	282.9546811
9	132	30	68	59.67222248	69.35187837
10	130	47	69	70.09411274	1.197082685
				SSE	594.6548125

Figure 3 Multiple Linear Regression using gradient descent

5. REFLECTIONS AND CONCLUSION

Computational thinking is best learnt and taught through doing and was carried out for the Specialist Diploma in Applied Artificial Intelligence where the students were taught Statistics using Computational Methods. At the end of the course, the students displayed competencies in using models to express problems and then using algorithmic method such as gradient descent to solve them. Students who were taught the previous methods of teaching statistics understood linear regression but found it hard to extend what they have learned to multiple linear regression. Unfortunately, as the number of independent variables increases the calculations to get the predicted equation relating Y the independent variable to the independent variables X becomes very complicated and there is no easy or simple way to generalize the equation for any number of independent variables X. The students were also unable to cope with the numerous formulas that they must learn, and the domain knowledge needed to apply these statistical formulas. Computational thinking avoids these shortcomings. The positive feedback from these students who were taught the Computational thinking method, gave the author the assurance that the framework so drawn has indeed addressed some aspects of computational thinking in statistics. The students were able to understand better the application of the regression method for linear and multiple linear regression compared to teaching them the traditional statistical approach. The computational model based on the key ideas of abstraction, decomposition, algorithm thinking, pattern recognition and generalization has led to simple and convenient method of finding the general linear equation and can be automated using MS-Excel or any other programming languages. Going forward, the aim is to approach the research design with more quantitative measures to ascertain if the learning outcomes attained are significant.

6. REFERENCES

Hinton, P. R. (2014). *Statistics Explained Third Edition Book.* London and New York: Routledge, pp. 244-272, pp. 294-315

Looi, C.K. (2017), Computational Thinking for Every Student, InfoComm Media Development Authority (IMDA), Impact News, Retrieved November 13, 2017 from https://www.imda.gov.sg/news-and-events/impact-news/2017/11/computational-thinking-for-every-student

Wing, J. M. (2006). Computational Thinking. Communications of the ACM, 49(3), 33-35.

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

運算思維模組化教學活動設計：幾何之美

楊心淵 [1]，許庭嘉 [2*]，溫韋妮 [3]

[1] 嘉義市北興國民中學，臺灣

[2, 3] 國立臺灣師範大學科技應用與人力資源發展學系，臺灣

tnjboxing@gmail.com，ckhsu@ntnu.edu.tw，roru0945@gmail.com

摘要

模組化在程式設計中是相對困難的概念，學生很難理 解與應用。本課程以四個任務活動，讓學生在任務挑 戰歷程中，完成模組化程式概念學習及應用。本次模 組化運算思維教學設計總共七節進行四個子活動，活 動一以歌詞模組化學習模組化概念，活動二 Code.org 藝術家 4 進行線上闖關，從函式呼叫到函式建立，螺旋漸進學習函式的概念及積木程式撰寫，活動三正多邊 形繪製則讓學生學習自訂函式積木，並運用函式積木 完成任務，最後活動四則以自然環境規律之美觀察， 並運用 Scratch 模組化積木電腦繪製自然之美意象，除了程式最終作品 老師評分外，也包括作品發表及同儕 之間的自評與互評。

關鍵字

運算思維；Scratch；模組化；積木程式

1. 前言

本單元課程目標聚焦於八年級模組化概念及程式設計 教學，以自然規律之美幾何繪圖為表現任務，學生觀 察自然規律之美圖騰，找出自然規律之美作業圖片， 運用運算思維解析圖形，拆解找出最小重複單元，樣 式辨識找出圖形之規律與脈絡，並透過 Scratch 進行程式繪圖實作，以電腦幾何繪圖模擬展現自然之美意 象。

要完成這個任務，學生須先具備模組化概念及程式實 作之能力，因此，以活動一『歌詞』先讓學生了解模 組化概念；活動二『code.org 藝術家 4』，以闖關方式強化模組化概念，並進行模組化程式實作練習；活動 三『多邊形繪製』以多邊形繪製任務，讓學生從函式 建立到函式應用，從函數、帶參數函數、巢狀函數到 函數綜合運用實作練習。完成活動一到活動三的基礎 能力學習， 最後以活動四自然規律之美，讓學生將前面所學模組化概念及程式設計綜合運用，完成表現任 務！

2. 文獻探討

2.1. *運算思維教學*

運算思維應採用跨領域的教學模式，深化跨領域的知識，並激發學生對科學、科技、工程及數學的興趣(Hsu, Chang, & Hung, 2018)。也就是說，當我們嘗試設計運算思維的課程時，有必要考慮如何融入其他領域 的知識內容，才能在達成預期學習目標的同時，提升 學生的學習動機，並深化學生對其他領域知識的整 合。

Durak 和 Saritepeci (2018)的研究指出：學生面對數學和科學課程的態度及學生在數學及科學學業成就上的表現，對學習運算思維的影響很大。因此，我們應優 先考量將數學或科學的知識融入運算思維教學的可能 性，期望透過數學或科學的融入，提升學生對數學或 科學的興趣，進而幫助他們學習運算思維。為了達到 上述的目的，此教案結合了數學的幾何概念，以及自 然科學中大自然之規律進行活動的設計。

2.2. *積木程式教學*

因應不同年齡段學生認知能力的不同，對於運算思維 所設計的教學內容及策略都應該因此而有所不同(Hsu et al., 2018)。Angeli 和 Giannakos (2020)也提到：學生學習運算思維時，有必要考慮為學生構築鷹架。在運算 思維的學習上，我們常採用程式撰寫的方式來進行教 學。然而，在學生認知能力還不足以獨立完成完整程 式撰寫時，我們需要考量提供額外的協助及教學以幫 助學生完成學習目標。

我們應在越來越多的學生友善型的程式撰寫環境(Alice、Scartch 等)、硬體(3D 列印、教育機器人)和其他線上的程式教學平台(如 code.org、codeacademy.com) 出現的情況下， 考量該如何進行運算思維的教學

(Angeli & Giannakos, 2020)。因此，結合前段所提到的，考量到八年級學生尚未具有獨自撰寫程式的能力，決定使用 Scratch 積木程式及 Code.org 藝術家 4 圖形化程式學習網站，在簡化、模組化下的積木程式撰 寫環境中，帶領學生透過程式撰寫，學習運算思維。

3. 教學設計

3.1. 教學對象

本課程之教學對象為嘉義市某國中八年級之學生，二 個班級共 58 位同學參加此運算思維課程，共計四個活動單元，最後教學生熟悉 Scratch 積木程式模組化實作。

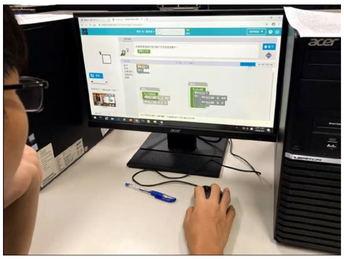


圖 1 課堂中學生操作 Scratch 積木程式

3.2. 教學活動設計

本次教學設計總共包括四個活動，在台灣的中學一節課為 45 分鐘，活動一需花 1 節課，活動二需要 2 節課，活動三需花 2 節課，活動四需要 2 節課，總共七節課完成模組化學習，教學設計如下圖 2 所示。

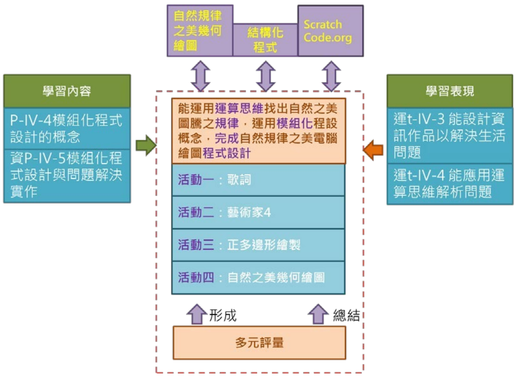


圖 2 運算思維素養導向教學設計模式教學模式

活動一是以學生喜愛當紅歌曲的歌詞，模擬電腦程式 碼執行，透過找尋歌曲重複的地方來定義函數，比較 歌曲函數定義前後的差別，藉此讓學生對函數有基本 的認識及概念。

活動二是 Code.org 藝術家 4，如下圖 3 所示。學習內容從認識函數、定義、操作函數、函數綜合運用為課程核心，前面單元，學生已經對函數有基本的概念，本 單元則透過 Code.org 藝術家 4，以螺旋漸進的函數學習任務關卡，除了提升學生函數概念外，也讓學生練習 函數程式修改練習實作。

圖 3 Code.org 藝術家:Level9 自定義帶參數巢狀函數

活動一、二已讓學生對於函數概念及程式實作有基本 概念，本單元結合了數學的幾何概念，以正多邊形繪 製為學習載體，除了以結構化程式複習七年級結構化 程式概念外，也運用其結果定義為函數，並實際運用 多邊形函數完成單元任務。

活動四以自然規律之美幾何繪圖為表現任務，學生須 運用前面所學模組化概念及程式設計技能，並透過自 然科學的觀察找出生活中自然規律之美及其規律與脈 絡， 並透過 Scratch 進行程式繪圖實作，以電腦幾何繪圖模擬展現自然之美的意象。

3.3. 教學評量

本課程透過學習單、Code.org 藝術家 4、Scratch 程式實作和學生自評互評表來進行學習評量，如下表 1 所示。最後由老師評定學生 Scratch 實作之完成度，以等第方式來表示。

表 1 本課程之自評及互評表

自然規律之美程式繪圖–學生自評互評表

自評者： 座號：　　　姓名：	完全符合	部分符合	不符合	備註
原始圖片是否具有重複規律之美				
所寫程式運用模組化程式(函數)				
所寫程式運用帶參數函數				
所寫程式運用巢狀函數				
繪圖結果能表達原始圖片規律之美意象				
能清楚說明圖形重複的單元及規則脈絡				

受評者一： 座號：　　　姓名：	完全符合	部分符合	不符合	備註
原始圖片是否具有重複規律之美				
所寫程式運用模組化程式(函數)				
所寫程式運用帶參數函數				
所寫程式運用巢狀函數				
繪圖結果能表達原始圖片規律之美意象				
能清楚說明圖形重複的單元及規則脈絡				

4. 結果與討論

4.1. 學生學習結果

活動一讓學生能運用運算思維解析歌詞，將歌詞以模 組化方式呈現，並能舉出幾個模組化在生活中的應用 例子，學生成果的實例如下圖範例所示。

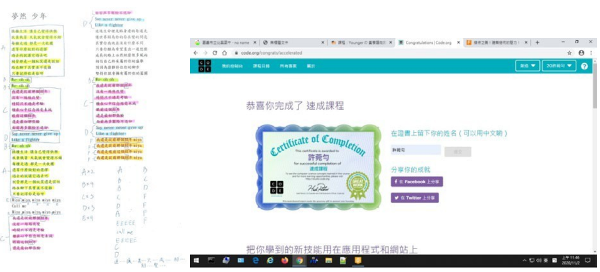


圖 4 活動一(左圖)與活動二(右圖)學生學習成果示例

活動三讓學生能運用運算思維找出多邊形繪製之規律，設計模組化程式，完成多邊形幾何繪圖之任務。然後在活動四規劃安排學生分組，並以仿世界咖啡館 方式，各人作業，分組發表方式，進行個人作品發表 展示，如圖 5 所示，最後並進行學生間的自評與互評。

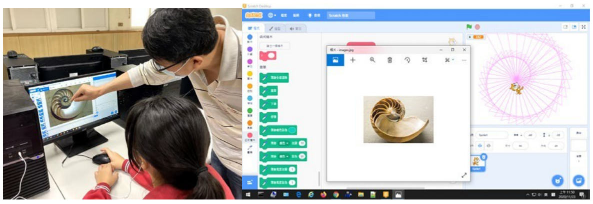


圖 5 本課程學生實作範例

活動四結束後，學生的 Scratch 實作成果由教師評估其完成度，總共分成十個等第的完成度，最差為 D，最優為 A+，人數分配形成雙峰現象，如表 2 所示。

表 2 最終積木程式實作之各等第完成度人數分配

等第	D	C-	C	C+	B-	B	B+	A-	A	A+
人數	2	4	7	8	1	4	4	7	10	11

4.2. 教師反思

從活動一到活動三模組化概念及程式設計基礎練習、 到活動四模組化概念綜合應用，教師應注意引導學生 問題解析，回顧活動一到活動三學習的內涵，讓學生 可以學習遷移，將前面所學概念，應用在活動四。

本案為教學方便，以自然規律之美及幾何繪圖為學習 素材，教師應引導學生思考，舉一反三概念遷移，找 出模組化在日常生活中的各種運用實例，也思考模組 化還可以應用在生活哪些地方？

活動四採個人作業，分組發表方式，可讓學生每個人 都有發表的機會，並能有效掌控時間！學生也因為自 評、互評表清楚知道評量方式，更能讓學習聚焦。

5. 結論與建議

本課程首先以歌詞作為模組化概念的引導，接著以 Code.org 藝術家 4、Scratch 程式實作讓學生了解積木程式的撰寫方式，最後透過 Scratch 程式繪圖實作，將自然之美的意象以電腦幾何繪圖模擬呈現出來。過程中 除了實作評量，也透過學習單進行形成性評量，最後 的學生自評互評表，則讓學生評估自我學習成果，並 學習從他人的成果中學習他人的優點以精進自己。學 生在學習過程中能夠發現生活中如歌詞、幾何圖形中 都存在著規律，並能夠運用程式以模組化的方式找出 其中的規律。

對於未來教師發展運算思維教學活動，以下有幾點建 議期能提供設計時之參考：

(1) 運算思維為較抽象的概念，可以貼近學生生活經驗 之素材進行考量，學生較能體會其中的內涵。如本 課程中使用歌詞及幾何圖形來引導學生學習模組化 概念。

(2) 進行 Scratch 程式教學時，應視學生能力適時調整對學生的引導。並要注意對過去所學概念的連結， 讓學生能夠學習遷移。

(3) 跨領域知識可與其他領域的教師協同教學，如此教 學活動中的幾何圖形，可由數學教師教導學生了解 各種幾何圖形的定義；而由規律圖形所呈現之自然 之美(對稱、比例等)，則可以與美術老師合作，由 美術老師來進行相關的教學。

6. 致謝

本課程榮獲臺灣教育部國教署 2020 年國中組資訊科技教案設計競賽獎勵並出版教案供全國中學教學參考。 本研究感謝科技部研究計畫編號：108-2511-H-003 -056 -MY3 補助。

7. 參考文獻

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*, 106185. https://doi.org/10.1016/j.chb.2019.106185

Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116*, 191-202. https://doi.org/10.1016/j.compedu.2017.09.004

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Modeling Instruction Design for Computational Thinking Activities: Geometric Beauty

Hsin-Yuan YANG[1], Ting-Chia HSU[2*], Wei-Ni WEN[3]
[1] Chiayi Municipal Beixing Junior High School, Taiwan
[2, 3]National Taiwan Normal University, Taiwan
tnjboxing@gmail.com, ckhsu@ntnu.edu.tw, roru0945@gmail.com

ABSTRACT

Modeling is a relatively difficult concept in programming, which is hard for the students to understand and apply. This course develops a 7-period instructional material integrating four activities for modeling instruction in computational thinking activities. The first activity asked the students to pick out the repeated lyric from their daily-life songs, so as to process the repeated parts in one time. The students recorded their learning process with a worksheet for this activity. The second activity asked the students to complete the games of the artist 4 in Code.org. After the students passed all the requirements, they got the certificate from the website of Code.org. The students had to draw the polygon with Scratch programming design in the third activity. Finally, the students had to apply their observation of the pictures from their daily-life in the nature to implementing the geometry. The final block-based programming creation was evaluated by the students themselves and their peers. Finally, the teacher also assessed the completeness degree of their implementation.

KEYWORDS

Computational thinking, Scratch, Modeling, Block-based programming

Computational Thinking and Artificial Intelligence Education

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

運算思維教育的教學反思:
運用運算思維結合人工智能提升學生的創意解難能力

陳景康 [1*], 許文星 [2*], 賴家豪 [3*]

[1,2,3] 英皇書院同學會小學第二校，香港

khchan@kcobaps2.edu.hk, mshsu@kcobaps2.edu.hk, khlai@kcobaps2.edu.hk

摘要

本文旨在闡述 3 名年約 10-11 歲的小學生如何運用運算思維結合簡單的人工智能概念，創作「抗疫次廁安心」智能座廁裝置來解決新冠肺炎疫情下的一個公共衛生問題。老師在課堂中引導學生利用運算思維解決問題技巧（Practices）思考整個流程的步驟以及整個裝置的功能和構造，配合 MIT Scratch3.0 編寫具人工智能概念的程式，同時運用 LEGO EV3 各種感應器建構模型，幫助學生將構思實踐出來。本文的教學實踐闡明人工智能元素的運用在運算思維教育的可行性，下文會以「抗疫次廁安心」智能座廁裝置作為例子，並進行探討。

關鍵字

運算思維；編程；Scratch3.0；人工智能；零接觸

1. 前言

運算思維（Computational Thinking），又稱計算思維，所謂運算思維就是「利用電腦科學的基本概念進行問題解決、系統設計與人類行為理解的思維模式。」（Jeannette M. Wing，2006）其目的是培養學生運用數碼創意，提升解難能力，並且能學以致用，協助解決社區問題。故此教育局課程發展處建議在小學階段引入編程來培養學生的運算思維，希望通過適當設計的學習活動，為學生提供獲取和應用運算思維和編程技巧的機會（課程發展議會， 2017）。而人工智能 （AI） 是指可模仿人類智能來執行任務，並基於收集的信息對自身進行迭代式改進的系統和機器。事實上，AI 已逐步融入我們的日常生活，而且技術亦日趨成熟，意味下一代的未來，將與 AI 密不可分。本文將探討如何在運算思維教育中引導學生結合人工智能概念，擴闊數碼創意的空間，更有效解決社區問題。

2. 創作過程

2.1 發掘及分析問題

運算思維教育重視培養學生提出問題和利用編程的相關技能來解決社區問題，因此，教師在日常教學中常常鼓勵學生關心身邊事物及新聞時事，如透過剪報及新聞分享等的方式，加強學生對社會問題的認識。

去年至今，新冠肺炎持續肆虐全球，世界各地無數人深受其害，對學生的影響亦是前所未見。故此學生認為最想解決與衛生相關的社區問題。

學生留意到由於此病毒主要透過飛沫和接觸傳播，而患者的排泄物亦可能充滿病毒，所以衛生署建議市民上完洗手間後要先把廁板蓋好才沖廁，避免病毒隨處飄散。而這樣的傳播病毒危機在公眾地方的洗手間尤其嚴重，因為這些地方的使用頻率比較高，加上每個人的如廁習慣及衛生意識不同，這些因素都容易造成衛生問題，甚至可能導致疫情爆發。

學生先拆解面對的問題。透過資料蒐集及訪問學校洗手間使用者，歸納出透過公共洗手間的座廁傳播病毒的途徑，主要包括:

1. 人們沖廁時忘記蓋上廁板，令病毒隨處飄散；
2. 使用者動手開合廁板時容易污染雙手；
3. 使用者打開廁板才發覺座廁未清潔乾淨。

2.2 大膽假設，發揮創意

基於以上分析，學生假設「接觸廁所愈少，愈可以減低感染病毒的機會」，從而提出「零接觸」的意念。而學生認 為只要做到以下三點，就能做到「零接觸」：

1. 沖廁時廁板可以自動關上，避免沖廁不乾淨而導致病毒擴散的危機。
2. 座廁可以自動開啟，減少人們接觸廁所各裝置的需要，把接觸傳染風險減至最低。
3. 座廁具備自動清洗功能，並利用數據使其智能運作，可因應使用情況自動調整於閒置時進行自動清潔的頻 次。

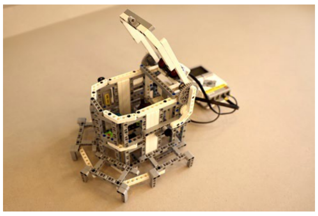
2.3 建構模型，實踐意念

為了實踐相關的意念，學生運用了運算思維中「重用

及整合」的概念，參考了市面上一些智能座廁的功能，如自動開合廁板、自動沖廁等，並利用 LEGO 積木及 LEGO EV3 建構座廁模型，再配合 Scratch3.0 編寫應用程式控制座廁模型上相應的 LEGO 感應器及組件：光線感測器、超聲波感測器及 Servo Motor；及顯示座廁的使用清況。

應用程式如何與裝置配合運作呢?這時，學生需要在編寫程式時，制定一套簡單的演算法則，以顯示明確的 規則和步驟，讓硬件和程式配合執行。首先，如果座 廁的超聲波感測器感應到有人行近，程式先檢視廁所 的清潔情況，如已經清潔乾淨，就會驅動 servo motor 自動打開廁板，而程式介面會顯示「occupied」，表示有人正在使用。

相反，當超聲波感測器偵測到有人離開，程式就會驅動servo motor，將廁板合上，並開始進行自動沖廁清潔。同時座廁內的光線感測器會透過偵測光線的反射情況，反覆檢查座廁內的清潔情況，決定是否需要再次沖廁，直至沖洗乾淨。而程式介面上會顯示「vacant」，表示座廁已經清潔乾淨，使用者可以安心使用。



「抗疫次『廁』安心」座廁模型

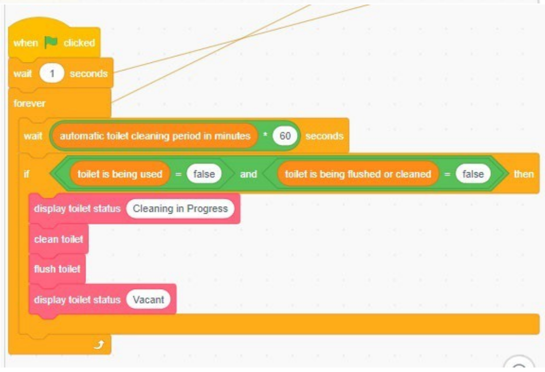
「抗疫次『廁』安心」程式介面

2.4 *應用人工智能完善方案*

為了解決公共洗手間多人使用而造成的衛生問題，學生們特別為座廁加入了自動清洗功能，令使用者更安心使用座廁。但討論過程中，學生發現如果只在程式中加入時間設定，在實際操作時可能會發生以下問題：

1. 若設定時間間隔太短，座廁啟動自動清洗時，可能 影響使用者；
2. 若設定時間間隔太長，可能影響座廁的清潔情況。

為了解決以上的問題，學生與老師討論後，便嘗試為程式加入人工智能元素，利用數據讓程式自行判斷清潔的頻次，解決以上問題。方法如下：程式會每天讀取前一天每小時座廁的使用量次數，得出自動清洗廁所的時間段。使用次數按小時記錄。使用次數越多，之後一天該時段的廁所清洗的頻率就越高，令所有人使用廁所更加安心。



3. 反思

完成此裝置及相關程式後，老師鼓勵同學對作品進行反思。同學們都覺得「抗疫次廁安心」 智能座廁這個設計， 特別是人工智能的應用，還有很多可以改善的地方。例如學生在測試程式時發現他們設計的人工智能自動清潔功能在不同使用環境可能會出現問題，因為程式是以廁所前一天的使用量決定自動清潔的頻率，但在學校使用時，假期時廁所的使用率會大幅減少，但到了上學日，使用人數增加，人工智能決定的清潔次數就會出現偏差。然而，運算思維教育並不是要求學生能設計出一個完美的程式， 相反，如前所述，運算思維教育的其中一個目的是透過拆解問題及算法思 維等培養學生解決問題的能力，在設計程式過程中， 學生不但學會了拆解問題的技巧， 更重要的是養成 「反覆構思及漸進編程」的態度， 他們不斷審視每個可見的問題並進行除錯及提出方案。

4. 總結

美國麻省理工學院媒體實驗室媒體藝術與科學副教授

Cynthia Breazeal 博士在「學與教博覽 2019」一個講座中提到，學生在幼稚園階段就應該開始接觸人工智能 及學習運算思維，關鍵在於使用合適的工具和方法。 雖然「抗疫次廁安心」智能座廁在人工智能的應用上 只是一個初步的嘗試，然而，這次經驗不但證明了小 學生有能力掌握人工智能的知識和概念，更可以擴闊 了學生設計應用程式時的創意空間，培養他們的解決 社區問題的能力和動力，真正體會數碼充權的意義。

5. 參考文獻

Wing, J. (2006). Computational Thinking. Communicationsof the ACM, 49(3), 33–35. doi:10.1145/1118178.1118215

課程發展議會(2017)。《計算思維─編程教育：小學課程補充文件》。香港：課程發展議會

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Using Computational Thinking Combined with Artificial Intelligence to Enhance Students' Creative Problem-Solving Ability

King Hong Chan[1*], Man Sing Hsu[2*], Ka Ho Lai[3*]
[1,2,3] King's College Old Boys' Association Primary School No.2, Hong Kong
khchan@kcobaps2.edu.hk， mshsu@kcobaps2.edu.hk， khlai@kcobaps2.edu.hk

ABSTRACT

This article aims to explain how three primary school students aged about 10-11 used computational thinking combined with simple artificial intelligence concepts to create a "tata germ, toilet" smart toilet device to solve a public health problem under the COVID-19 epidemic. In the classroom, the teacher guides students to use computational thinking problem-solving skills (Practices) to think about the steps of the entire process and the function and structure of the entire device, compile programs with artificial intelligence concepts with MIT Scratch3.0, and use LEGO EV3 sensors to construct models, to help students put the idea into practice. The teaching practice in this article clarifies the feasibility of using artificial intelligence elements in computational thinking education. The following will take the " tata germs, toilet “smart toilet device as an example and discuss it.

KEYWORDS

Computational thinking, Scratch3.0, Artificial Intelligence, No-touch

Computational Thinking and Evaluation

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Upscaling Skills-Based Formative Assessment: The Journey Towards a Student-Run Web Application Pilot on Computational Thinking Skills

Aaron HO [1], Yu Jie NG[2*]
[1,2]Dunman High School, Singapore
ho.jiawei.aaron@dhs.sg / ho_jiawei_aaron@moe.edu.sg, ng.yujie@dhs.sg

ABSTRACT

An investigation on an in-house Computational Thinking (CT) assessment method led to the development of the CT Quest web application pilot to mitigate upscaling challenges of rolling out a CT skills-based curriculum. The investigation showed that the more easily implementable quiz-based CT assessment method is insufficient to determine a student's CT ability accurately. The method may complement but not replace the Evidenced-Centered Design (ECD) CT assessment method used. This paper also outlines how a school can develop and customize CT assessment rubrics for K-12 students and the potential for non-computing subjects as well.

KEYWORDS

Computational Thinking, Formative Assessment for Learning, K-12, Computing, Cattel-Horn-Carroll (CHC) model of intelligence

1. BACKGROUND AND MOTIVATION

The school's Junior High 2-year programme includes coding (with Python), Data Science and Machine Learning tools for this era of Artificial Intelligence. In Senior High, students taking the 2-year Higher 2 Computing GCE A-Level examination deepen their knowledge, interact with data, develop and apply suitable algorithms and data structures to solve real-world problems as well as participate in various international competitions. Thus, there is a great need to quickly immerse students in CT to deconstruct problems with confidence, use datasets to communicate complex ideas with technological tools with more clarity, propose user-centric solutions and apply metacognitive skills within iterative hands-on product development experiences.

To achieve these aims, teachers seek to develop students' willingness, competencies and intuition to break down complex problems into more manageable parts (decomposition), recognise common patterns (pattern recognition), identify and model essential components (abstraction), devise well defined procedures (algorithm design) and translate these to machine processable constructs (programming) to derive the benefits of automation to enhance their productivity and quality of lives and those of others. CT skills in non-coding areas such as User Interface and User Experience (UI/UX) and infographic design are also taught and assessed with the aim of developing students' intuition in CT's transferability (Wing, 2006). In addition to the typical four CT skills mentioned above, two more CT skills are assessed for a more complete evaluation of the CT process: "Metacognition" and "Learning Behaviours" which are "using self-reflection to regulate and assess usage of the above 4 CT skills" and "necessary approaches, habits and strategies utilised during CT: resilience, resourcefulness, creativity, communication, disaffection, responsibility, collaboration & reciprocity" respectively (Allsop, 2019).

Therefore, there is great interest from teachers in developing CT intuition and habits.

2. METHOD

Teachers wanted to establish a credible and implementable CT assessment method suitable for the school's student profile before rolling out a CT curriculum to all Computing lessons and levels. This assessment method must be scalable and should aim to capture, monitor and assess CT in the thinking processes of students over time without overwhelming teachers. So, the investigation set out to evaluate if the design of questions to elicit CT written responses within a quiz-based assessment could meet this need. If successful, this methodology for question design would prove reliable and give more credibility and meaning for upscaling - building more systems such as online platforms to facilitate the capturing and processing of students' responses to these questions for effective CT teaching and learning.

Therefore, to test if the question design and grading processes in the quiz-based assessments were able to capture an accurate representation of each student's CT ability, the students' scores of a quiz-based assessment and an ECD assessment were compared.

The **quiz-based assessments** were Kahoot MCQ quizzes developed using combinations and adaptations of existing CT pedagogies and literature which will be described in detail below. Besides typical CT definition and basic CT concept questions, students were also required to evaluate and rank descriptions of CT in other students' work on the same type of project they were working on. These descriptions were aligned to the CT assessment rubric level descriptors which will be described in detail below.

In contrast, the **ECD assessment** was known to the school's teachers to be a tedious, unsustainable but reliable way for teachers to establish a score for each student's "true" CT ability because of its extensiveness in documenting and quantifying students' thinking processes and work. It served as a comparison standard for the quiz-based assessment. To obtain each student's "true" CT ability, two Computing projects were selected to be graded by ECD assessment: A whole cohort of Year 1s (13-year-olds) and Year 2s (14-year-olds) used web-based applications Piktochart and Thunkable to create an infographic and build a prototype of a mobile application respectively.

The CT assessment rubric features 3 - 4 level descriptors for each of the 6 CT skills. Descriptors were developed through selecting and categorising teachers' observations of various degrees good and poor applications of CT from past student batches' work based on the ECD provided by SRI International's Principled Assessment of CT (Bienkowski, Snow, Rutstein, & Grover, 2015) and the faculty and administrators at the University of Delaware Center for Teaching and Assessment of Learning (University of Delaware, 2021). Teachers then contextualised it for the project assessed (i.e. Piktochart infographic or Thunkable mobile application) and rephrased it for simplicity and clarity.

In alignment with this rubric, the methods used for observations, evaluations (ECD assessment) and designing and grading questions (quiz-based assessment) were guided by the work of Marcos Román-González, Juan-Carlos Pérez-González, and Carmen Jiménez-Fernández (2017) and Brennan, K., & Resnick, M. (2012) by measuring CT through the CHC model of intelligence (i.e. fluid and crystallised intelligence and three-stratum hierarchy) in project portfolio analysis, artifact-based reflections, and design scenarios.

For the ECD assessment, students developed their products over a few months and submitted it with pieces of evidence of CT in the form of write ups, rough work and even email conversations. For the writeups, students were provided with guiding questions with hints on the type and aspects of CT skills assessed. (See Annex for more question examples)

Four Teachers did a standardisation exercise by grading a few students' work and then finalizing on a grade to which other gradings would take reference from. This helped reduce the subjectivity in grading.

During the ECD assessment grading process, teachers first read the title of the students' app and used the app as an end-user to come up with a preliminary grade for the CT that might have been needed to be shown to achieve the current complexity and quality. Next, the teacher would read the write-up and supporting documents to better understand the process and extent of CT used and may then adjust the grade. Evaluation of the student's thought processes included analysing the quality of the questions (Brooks, 2019) asked by the students in their responses (students were asked to share the questions they asked in their thought processes. See Annex for more question examples). For the Thunkable project which was done by a group of students, the teacher would also cross-check same team members' writeups (each member writes independently) to get a better picture of CT skills shown in the planning, collaboration and accuracy of the writeups.

3. RESULTS

Figure 1 shows the range of the number of students who had very little difference in scores (expressed as a percentage for comparison purposes) to the number of students who had very big differences in scores. This difference is the score for the quiz-based assessment subtracted from the score of the ECD assessment. With an average of -1.97%, more students obtain lower scores in the ECD assessment. The r-values for the Piktochart quiz (0.06) and Thunkable quiz (0.07) are each less than 0.7 thus indicating insignificant correlation of both quizzes with ECD assessment scores; coupled with the scatter-plot diagrams, the quiz-based scores hardly explain ECD assessment scores.

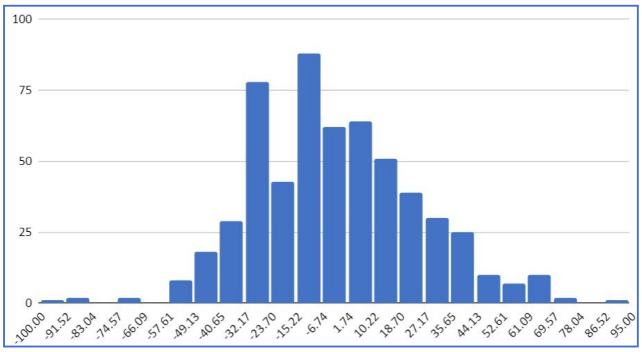


Figure 1. Number of students against the difference of scores (in terms of percentages for comparison's sake) seen in ECD assessment as compared to quiz-based assessment.

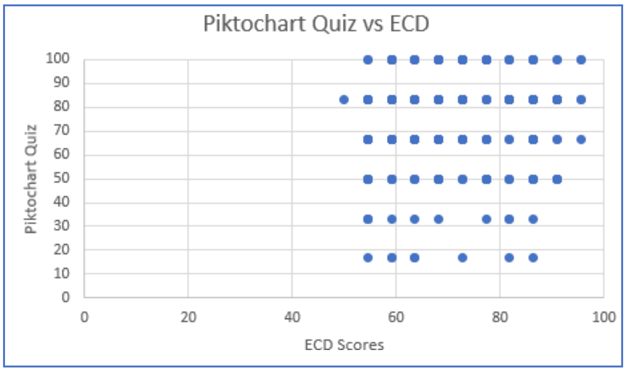


Figure 2. Scatter-plot, PiktochartQuiz vs ECD Scores

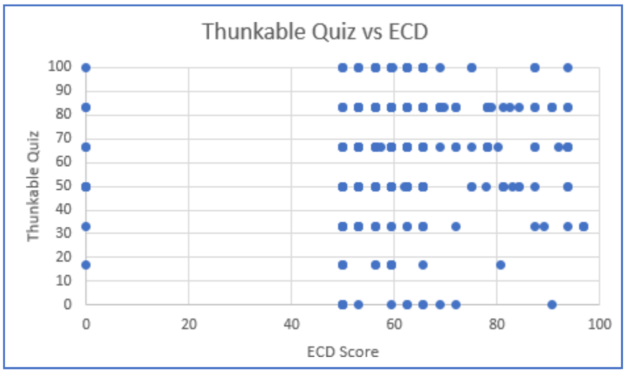


Figure 3. Scatter-plot, Thunkable Quiz vs ECDScores

4. DISCUSSION

There is insignificant correlation of student performance between the quiz-based and ECD CT assessments. So, there is insufficient ground to conclude that the method used to design the quiz-based assessments can represent a student's "true" CT ability. Based on the above results, more than 25.8% of students could have been scored inaccurately by a difference of a grade or more. So, more research and development may be needed to improve the quiz-based assessment, be it design, frequency and/or grading methods. Therefore, ECD CT assessments cannot be replaced entirely by the current quality of quiz-based CT assessments.

A significant assumption made was that the ECD assessment was a reliable way for teachers to know the students' "true" CT ability. This was however taken as a reasonable assumption by the teachers as it was the best effort at their manpower capacity at the point of investigation.

Less students seemed to do better in the ECD assessment than the quiz-based assessment. Teachers attribute this to the more encompassing and rigorous nature of ECD assessment and its better ability to evaluate application of CT. However, the accuracy of the quiz-based assessment may improve with more questions and frequency that the test was conducted (only four were conducted).

The students who improved the most in ECD assessments revealed that the time limit and pressure of doing well for the quiz-based assessment seemed to inhibit them from performing better. In-class observations validated that these students really do reflect a deeper level of problem-solving skills, have prior computing experience, or have cognitive resilience and have spent much time on their project outside of school. As for the students and outliers who showed the greatest deterioration of scores, a majority was due to discipline or personal challenges not related to the CT tasks.

Teachers' and students' feedback on the above process indicate awareness of the great need and potential of CT for the jobs of the future as well as the interest to develop in CT but lack of a platform to facilitate the formative aspects of CT for the learning of CT to be more effective and efficient.

Even if the current ECD assessment was taken to be the method to represent CT ability, its current implementation is too challenging for an entire curriculum's assessment. Teachers are aware that CT is best evaluated holistically, continuously and personally; ideally, teachers want to assess and track students individually over various mentorship sessions for a more holistic and fair assessment. However, this is unfeasible in many typical school logistical and curriculum limitations. Capturing and monitoring multiple email conversations, collating project artefacts and grading write-ups is overwhelming, messy and could deter teachers from doing CT altogether. Such grading methods also favours the eloquent or interactive students who engage teachers more. Furthermore, the CT thought process is often forgotten by the time the student writes it down in the write ups. As for the students, they wished for more explicit scaffolding and prompts to use CT while they are in the middle of problem solving and routines to make CT visible and rewarded. Also, students wished for more alignment to the curriculum and syllabus so that they could see explicitly how CT helps them improve during revision and in their final examinations.

5. FOLLOW UP

To address the above issues collectively, a prototype web-application "CT Quest" is conceptualized and built with Year 6 Computing students (Team leads: Leo Qiyi Joel and Wang Yaohui. Team members: Isaac Chen Jing De, Kingold Wang, Liu Hongshuo and Ng Jia Xiang) as part of a bigger school app. CT Quest applies the earlier investigation's results by focusing more on ECD assessment features than on quiz-based assessment methods. It is designed with the potential to be used for other subjects in the future as well and can be particularly helpful for scaling skill-based learning within communities and group revision for examinations.

CT Quest lets teachers track and reward students with points for "tagging" parts of their answer and explaining the method used. This "tagging" can also be done for projects, and conversations with teachers via the platform's integrated messaging feature.

A student can quickly identify learning gaps (i.e. lack of display of a certain skill or a content knowledge of a curriculum topic) by sorting and filtering search results of all his/her work done over the year, so that he or she can see what is lacking from a tabular or listed display of ranked work attempts. So, each attempt (i.e. a question marked, a correction done, or a conversation with the teacher) can and should be "tagged" to a skill and topic category from the curriculum for this sorting, filtering and ranking feature to work well.

Students can click on each skill to view an assessment rubric on how the skill is assessed. Different levels of display of the skill are described with examples and points scored. This helps students know the requirements and how to improve (e.g.: how to improve a level 2 display of a skill to level 4). Also, teachers would have an interface to adjust and update the rubric anytime.

With a click, students can see an overview of the sum of points earned for each skill and topic so that they can identify the skills/topics they are strongest and weakest in. Students can click on each skill/topic to display a list of questions which had contributed to that sum of points. Students can then click on each question to "zoom in" to see the question content, as well as any related answers, comments and corrections done on it. After a student has gotten his/her answers marked, he/she will have an option to make corrections to his/her answers. While students can change their answers after submission, markers and teachers will be able to see earlier versions of their answers. Students would be motivated by frequent displays of a leader board showing top ranked classmates with the highest points, as well as the rewards they can redeem for each achievement level tier they reach.

CT Quest also detects and offers games to students who may be stuck while attempting questions or who may want to hone their skills further and earn CT points through playing the game. These games help students learn the skills in a more explicit and memorable way, as well as provide a form of relief, break or even rescue from a mental block or giving up attempting the question. The games can be tagged to existing question and answer content where relevant.

The purpose of this game capability in CT Quest is to grow two student-teacher communities. One uses these games to teach and learn while the other builds the games. Together, they enable teachers to select and customise games for the questions they are setting with ease.

When a question is being set, the setter determines how the skills will be displayed for the person attempting to fill in answers. This involves deciding if there should be an option provided for the person attempting to tag parts of their own answer with a skill from the skills list, to provide an accompanying description of how the skill was used to achieve the answer, and/or to write a descriptive reflection on the whole question on the method used.

These features address the challenges of ECD assessment by facilitating documentation and tracking of skills learned via more intentional collection of CT evidence in context of gamification and student-friendly, student-originated UX designs.

6. CONCLUSION

While the CT Quest solution to teach CT effectively and efficiently is yet to be validated, the earlier investigation was sufficient to measure the effectiveness of a quiz-based CT assessment method and conclude that ECD assessment should still be the main assessment. The investigation also helped to elicit and clarify the needs of teachers and students to scope the requirements of a prototype app which would assist the roll out of CT at scale. As a next step, further evaluation of the effectiveness of quiz question types and ECD in CT Quest can serve to determine its suitability for the teaching and learning of CT.

7. ANNEX

As part of the CT assessment rubric, here are examples of questions categorized by CT skill that students are taught to ask as part of their CT process and grading.

Decomposition: Could this be broken down into more independent and interchangeable parts for different team members to work on separately? How do I simplify this? How did I decide what features the app should have? What is the main problem I am trying to solve? What subproblems did I break it down into? How did I ensure that my app is easy to use? Who are the main users of my app? What are some questions I had about software used to make apps? Is an app the most effective way to solve the problem I have chosen to solve?

Pattern Recognition: What patterns do I see? What comes next? What do these things have in common? Which one is the odd one out? Did I research other apps that are like this app? Could I have used existing built-in functions or libraries instead of coding from scratch?

Abstraction: What category does this belong to? What's the main idea? How would I group these? How can I represent this in a diagram, graph, timeline, map? How did I make my code shorter? How did I make the app's design component layout easier for my users? While coding, what did I do when I encountered errors - how can I make my code easier to debug?

Algorithmic Thinking: What are some instructions I can write for someone to repeat what I just did? Can I design a recipe or instruction manual for this? How can I write these steps more efficiently for me and for the person/computer who will process it?

Metacognition: What did I learn about CT today? What made it easy for me to do CT today? How did concept X help me understand concept Y? What other questions do I still have about CT at this stage? How can I do planning, evaluating, modifying, monitoring, reflecting better so I can do CT better? What plan did I make before I started building the app? How did I ensure that my plan was a good one? How well did I follow my plan?

Learning Behaviours: What challenges did I face in CT and how did I overcome them? What questions did I ask myself, my peers, or my teachers? What is most satisfying about CT and this project? How did working with my peers help? What different views from my group mates did I have? How did we work with the different views?

8. REFERENCES

Allsop (2019). *Assessing computational thinking process using a multiple evaluation approach*. International Journal of Child-Computer Interaction, Volume 19, 2019, Pages 30-55. Retrieved February 1 2021, from https://www.science direct.com/science/article/pii/S2212868918300588

Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look (SRI technical report).* Menlo Park, CA: SRI International. Retrieved February 1 2021, from http://pact .sri.com/resources.html

Brennan, K., & Resnick, M. (2012). *Using artifact-based interviews to study the development of computational thinking in interactive media design.* Retrieved February 1 2021, from https://web.media.mit.edu/~kbrennan/files/Bre nnan_Resnick_AERA2012_CT.pdf

Brooks, J. (2019). *Computational Thinking through Classroom Assessment*. Tech-Based Teaching: Computational Thinking in the Classroom. Retrieved February 1 2021, from https://medium.com/tech-based-teaching/computational-thinking-through-classroom-assess ment-6b8299ea0fb6

University of Delaware (2021). *COMPUTATIONAL THINKING RUBRIC*. Retrieved February 1 2021, from http s://cpb-us-w2.wpmucdn.com/sites.udel.edu/dist/4/8672/fil es/2018/12/Computational-Thinking-Rubric--2ktkkgv.pdf

Marcos Román-González, Juan-Carlos Pérez-González, Carmen Jiménez-Fernández (2017). *Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test*. Computers in Human Behavior, Volume 72, 2017, Pages 678-691, ISSN 0747-5632, via https://doi.org/10.1016/j.chb.2016.08.047. Retrieved February 1 2021, from https://www.sciencedirect.com/science/article/pii/S0747563216306185

Wing, J.M. (2006). *Computational Thinking*. Retrieved February 1 2021, from http://www.cs.cmu.edu/~./15110-s13/Wing06-ct.pdf

Computational Thinking and Non-formal Learning

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021*. Singapore: National Institute of Education.

From Computational Thinking to Computational Action with Arduino Programming Projects through Non-formal Learning

Poh-tin LEE[1]*, Chee-wah LOW[2]
[1,2]Bukit View Secondary School, Singapore
lee_poh_tin@moe.edu.sg, low_chee_wah@moe.edu.sg

ABSTRACT

This paper shares on the non-formal learning of Arduino Programming for students to develop Computational Thinking through project works which solve real-life problems pertaining to community and environment issues. The students are members of the school's Infocomm Club. They write program codes using the C Programming Language with integration to hardware sensors and actuators. The students translate their learning to computational action with the aim to solve community and environment problems by designing meaningful projects with a sense of purpose.

KEYWORDS

non-formal learning, programming, computational thinking, computational action, coding

1. INTRODUCTION

At the Bukit View Secondary School, junior members of the Infocomm Club learn Micro:bit and Scratch Programming at Secondary 1 level. Some senior members at Secondary 2 and 3 levels of age between 14 and 15 progress to learn Arduino Programming for projects on community and environment problems. This transformation of Computational Thinking (Wing, 2006) to Computational Action (Kafai, 2016; Tissenbaum et al., 2019) through participation in solving real-life problems is aligned with the Smart Nation goals (https://www.smartnation.gov.sg) of Singapore to support better living using technology.

2. PROCESS

The students work in project groups comprising of three members. They meet once a week for two months and learn through non-formal learning during after-school activities (Lee et al., 2019; Lee & Low, 2020). Each group brainstorms on their project ideas to solve real-life problems at school, home or community. Figure 1 shows the process leading to implementation of community and environment-based projects.

Figure 1. Process of Community and Environment Projects

3. LEARNING TO CODE IN ARDUINO PROGRAMMING

In these projects, the students use the Arduino microprocessor which is an open-source electronics platform based on easy-to-use hardware and software (www.arduino.cc). Arduino boards are able to read inputs through sensors and generate outputs such as turning on motors or alarms. The students learn C Programming Language using the Arduino IDE integrated development environment as shown in Figure 2.

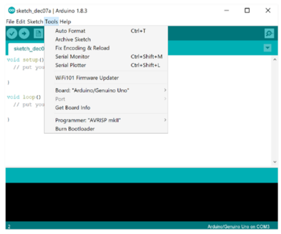


Figure 2. Arduino IDE development environment.

The senior student leaders in Infocomm Club assist to coach and mentor the groups in the development of the Arduino projects. Table 1 shows the software, hardware and integration skills learnt by the students.

Table 1. Software, Hardware and Integration skills learnt

Skills	Activity
Software	Write fundamental programming constructs using C Programming language at the Arduino IDE software environment.
Hardware	Connect the hardware sensors and output devices to Arduino microcontroller board.
Integration	Implement C Programs to read signal from or send signal to the hardware components such as sensors and actuators.

4. COMMUNITY AND ENVIRONMENT-BASED ARDUINO PROJECTS

These projects are designed such that students solve authentic real-world problems by leveraging on technology for issues pertaining to the community and environment. The students integrate skills on C Programming, electronic circuits and hardware sensors or actuators. For examples, Figure 3 and Figure 4 show the Fish Tank Monitoring Project and Maximise Solar Energy Project implemented.

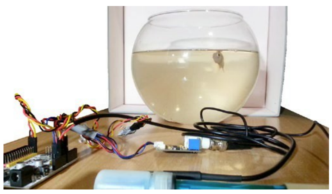


Figure 3. Fish Tank Monitoring Project

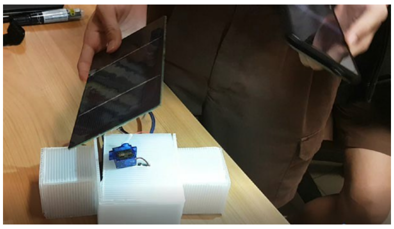


Figure 4. Maximise Solar Energy Project

Table 2 describes some of the community and environment-based Arduino projects implemented by the student groups.

Table 2. Community and Environment-based Projects

Project	Purpose	Description
Fish tank monitoring	Preservation of marine life	This project alerts fish owners on the water conditions of fish tank. It uses pH Meter, Temperature Sensor and Light Dependent Resistor (LDR) to measure pollution.
Maximise Solar Energy	Clean Energy	This project maximises solar energy charging by using Servo Motor to tilt the solar panel in an angle which follows the direction of sunlight.
Safe Classroom	Students Safety	This project minimises the risk of students getting hurt by using the Temperature Sensor and Humidity Sensor to detect fire.
Smart Switch	Energy Conservation	This project conserves energy by automatically switching off electrical devices. It uses Ultrasonic Sensors to detect the number of persons in the room.

5. CHALLENGES

Most of the students are more familiar with block-based languages in Micro:bit and Scratch coding. Hence, it is challenging for the students to learn the text-based C Programming Language (Weintrop & Wilensky, 2017). The following lists some problems encountered by the students in using text-based Programming Language.

a. *Syntax errors*: There were many syntax errors when the students' C Programs were initially compiled at Arduino IDE integrated development environment. Subsequently, the students learn that C Programming is a case-sensitive language and delimiter such as semi-colon cannot be omitted in the programs.

b. *Debugging*: The students spent much time to troubleshoot their program bugs as they did not understand error messages generated by Arduino IDE. They learn that many errors arise from incorrect conditional statements and connections of hardware sensors to pins on the Arduino microprocessor board.

6. CONCLUSION

This paper shares the process and implementation of Arduino Projects for the community and environment. The student groups work over a duration of two months through non-formal learning activities at the Infocomm Club. These projects enable the students to acquire skills in software coding and integration with hardware electronic sensors and actuators. Despite the challenges faced in learning the text-based C Programming Language, the students are motivated in creating these Arduino Projects as they find meaningfulness and a sense of purpose in developing solutions for real-world problems with a direct impact to the community and environment.

7. REFERENCES

Arduino (2021). *Why Arduino?* Retrieved from http://www.arduino.cc/en/Guide/Introduction/#why-arduino

Kafai, Y. B. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM, 59(8)*, 26-27.

Lee, P. T., Lee, X. R., Low, C. W., & Kokila, A. (2019). Implementing Computational Thinking through Non-formal Learning in after School Activities at Students Society Club. *Proceedings of the International Conference on Computational Thinking Education 2019*, 201-202.

Lee, P. T., & Low, C. W. (2020). Implementing Computational Thinking Curriculum with Robotic Coding Activities through Non-formal Learning. *Proceedings of the International Conference on Computational Thinking Education 2020*, 150-151.

Smart Nation (2018). *Smart Nation: The Way Forward.* Retrieved from https://www.smartnation.gov.sg/docs/default-source/default-document-library/smart-nation-strategy_nov2018.pdf?sfvrsn=3f5c2af8_2

Tissenbaum, M., Sheldon, J., & Ableson, H. (2019). From Computational Thinking to Computational Action. *Commnunications of the ACM, 62(3)*. 34-36.

Weintrop, D., & Wilensky, U. (2017). Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. *ACM Transasctions on Computing Education 18(1),* 3.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49(3)*, 33-35.

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Developing 21st Century Competencies and Computational Thinking through STEM-Based Co-Curricular Activities

Wei Sin HO[1*], Alex Han Rong YEO[2*], Lay Teng NEO[3*]
[1, 2,3] Maris Stella High School (Secondary), Singapore
ho_wei_sin@moe.edu.sg, yeo_han_rong_alex@moe.edu.sg, neo_lay_teng@moe.edu.sg

ABSTRACT

21st century competencies and computational thinking (CT) are viewed as essential skills for adapting and thriving in an increasingly globalised world. STEM-based activities offer opportunities for learners to develop these competencies and skills through hands-on learning and modelling solutions after real-world problems. This paper shares how 21st century competencies and computational thinking are developed through the co-curricular activities of a Singapore secondary school's Robotics and Programming Club.

KEYWORDS

21st century competencies, computational thinking, STEM, robotics, programming

1. INTRODUCTION

At Maris Stella High School (Secondary)'s Robotics and Programming Club, members between the ages of 12 and 16 participate in twice weekly sessions of co-curricular activities. These activities provide opportunities for the development of 21st century competencies (Ministry of Education Singapore, 2011) and computational thinking skills (Wing, 2006).

The club aims to enthuse students' passion about the possibilities of technological innovations through the following objectives:

- To foster a culture of effective communication, collaboration and student ownership. Students own their learning outcomes, as individuals or as part of a team.
- To challenge students to become critical and inventive thinkers through the embedding of the design process and computational thinking for innovation and problem solving.
- To nurture creativity and allow for creative expression of ideas and knowledge through challenge-based projects.

To provide breath exposure to current and emerging technologies, a range of customised programmes are conducted for members to hone their skills in the domains of cyber-security and cryptography, game development, HTML web programming, maker education, Python programming (Rashed & Ahsan, 2012) and 3D modelling.

To provide depth learning and acquisition of CT skills, its flagship programme is the offering of Lego Mindstorms EV3 robotics training.

2. THE ROBOTICS CURRICULUM

The Lego Mindstorms EV3 robotics curriculum is offered at three graduated levels of basic, intermediate and advanced to all lower secondary members with the aim of teaching basic programming and logical reasoning using robotics engineering contexts. Members work collaboratively in teams to complete activities and mini challenges. Table 1 shows the concepts and planned activities of the Robotics Basic Curriculum.

Table 1. Robotics Basic Curriculum.

Week	Concepts	Activities
1	Introduction: What is a robot?	Program a basic robot to go straight, curved, around a circle and a square.
2	Colour Sensor	Robot to follow a black line.
3	Medium Motor	Robot to collect and move a cuboid from one point to another.
4	Touch Sensor	Robot to navigate an obstacle course using touch sensor.
5	Gyro Sensor and Ultrasonic Sensor	Robot to move in a square formation using perfect 90 degrees turn.
6	Sound Display and Loop Blocks	Program robot to interpret two signals, and respond to the signals with a different behaviour.
7 - 8	Building Challenge: Gyro Boy and Colour Sorter	Build different robot models to investigate the use of the different sensors.
9	Building Challenge: Puppy	Build a puppy robot which will respond and react when students pet and feed it.
10	Robot Challenge: Robot Arm	Build robot arm to pick up objects in specific locations and deliver them.
11 - 12	Space Challenge	Build robot to collect the space commander in the fastest time.

3. DEVELOPMENT OF 21st CENTURY COMPETENCIES AND COMPUTATIONAL THINKING

Members learn to program the EV3 robots using the Lego Mindstorms Education EV3 software, installable on a computer or a mobile device. This interface allows students to code through block-based graphical programming. A block of program code using the Education EV3 software is shown in Figure 1.

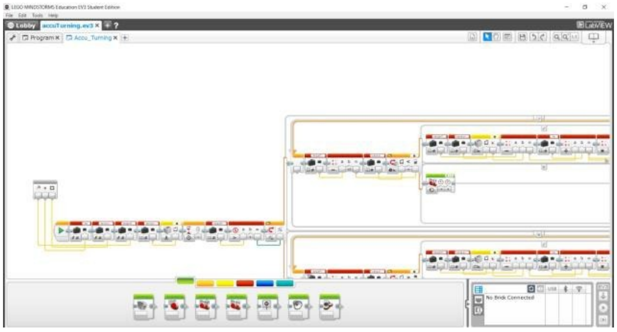


Figure 1. Program code showing an EV3 robot turning on the spot.

An example of a challenge is for teams to design and program one or more robots to pick up, transport and deposit up to twenty balls to various scoring panels around a field in a given time of three minutes. Figure 2 shows the final stage of the challenge.

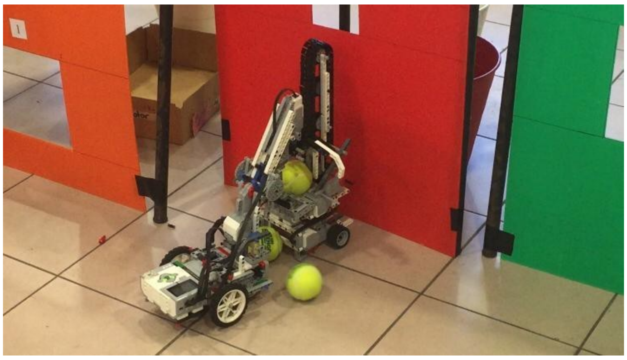


Figure 2. Students develop critical and inventive thinking as they encounter and develop solutions for real world applications and problems.

During the process of designing, constructing, and programming robots, students learn concepts of sequencing, branching, and loops. The performance of their robots demonstrate the outcome of students' computational practices. Students adopt a computational perspective by ensuring they are developing an understanding about the world and about themselves as producers and designers than just consumers of technology (Chalmers, 2018).

Results from various research studies has shown that educational robotics (ER) provides effective learning opportunities for the development of 21st century competencies such as creativity, collaboration, critical thinking, decision making, problem-solving and communication skills (Eguchi, 2014).

Table 2 below describes the CT skills developed at different stages of a robotics challenge, as well as the 21st century competency observed.

Table 2. Development of CT Skills and 21CC.

Problem Stage and CT Skill	21st Century Competencies
Students are given a challenge statement. In groups, students conduct scans, brainstorm for ideas and break down the whole problem into parts. **(DECOMPOSITION)**	Civic Literacy, Global Awareness and Cross-Cultural Skills
Students conduct research on current designs and models to identify patterns among and within parts. **(PATTERN RECOGNITION)**	Communication, Collaboration and Information Skills
They conceptualise, plan and design solutions. **(ALGORITHMIC THINKING)**	Critical and Inventive Thinking
Students build, code, test and refine their solution continually. **(EVALUATION)**	Resilience, Adaptability

4. CONCLUSION

STEM-based related activities such as robotics and game development provide hands-on applications of real-world problems, and are therefore instrumental in engaging students through nurturing of their digital competencies and promotion of creativity. Future studies could explore the impact of these activities on students' motivation and learning.

5. REFERENCES

Chalmers, C. (2018). Robotics and Computational Thinking in Primary School. *International Journal of Child Computer Interaction, 17,* 93-100.

Eguichi, A. (2014). Educational Robotics for Promoting 21st Century Skills. *Journal of Automation Mobile Robotics & Intelligent Systems, 8,* 5-11.

Ministry of Education, Singapore (2011). *Framework for21st Century Competencies and Student Outcomes.* Retrieved January 19, 2021, from https://www.moe.gov.sg/education-in-sg/21st-century-competencies.

Rashed, M. G., & Ahsan, R. (2012). Python in Computational Science: Applications and Possibilities. *International Journal of Computer Applications, 46(20)*, 26-30.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49(3)*, 33-35.

STEM Education Policies

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021*. Singapore: National Institute of Education.

中国西部地区 STEAM 与创客整合课程的现状调查与策略研究

贾越 [1]，陈梅 [2*]
[1，2] 内蒙古师范大学 教育学院，中国 内蒙古 呼和浩特
贾越@1005574467@qq.com，陈梅@nmchenmei@qq.com

摘要

在我国实施创新驱动发展战略的时代背景下，STEAM 教育与创客教育同为培养创新人才的有效途径。将 STEAM 教育与创客教育相结合能够更好的培养学生的计算思维与创新实践能力。本研究通过无结构访谈、实地调研和课堂观察等方法采集数据和资料，掌握中国西部地区 STEAM 与创客课程的发展现状，通过分析得出存在的问题和面临的挑战，有针对性的提出了五项策略与建议，以期推进国家西部地区 STEAM 与创客教育的发展。

关键词

中国西部地区；STEAM 教育；创客课程；现状研究

1. 现状调查

1.1. 调查设计与实施

目前，STEAM 视域下创客教育的研究主要关注国内经济发达地区，而聚焦于西部地区的 STEAM 与创客整合课程的研究较少，本研究对内蒙古自治区，宁夏自治区、 甘肃、新疆维吾尔自治区和青海等中国西部地区开展抽样调查研究，主要通过问卷调查与无结构访谈在广度上掌握西部地区 STEAM 与创客课程开展的基本情况。通过实地调研和课堂观察的方法获取第一手资料，在深度上把握西部地区 STEAM 与创客教育的发展现状，并且为了兼顾二者，本研究在调查创客教育方面主要采用了实地调查，在 STEAM 教育方面主要采用了课堂观察。但由于篇幅限制，笔者不在此赘述具体过程。

1.2. 调查数据分析

1.2.1. 基本情况

随着 STEAM 与创客教育的兴起与普及，我国大部分地区已经较好的开展了 STEAM 与创客整合课程，但对于较偏远的中国西部地区，STEAM 与创客整合课程还没有很好的普及以及更加深入的研究。通过调查研究发现中国西部地区在整合课程中 STEAM 的理念融入创客教育中的应用体现不足，缺乏推进区域发展的顶层设计， 各地区发展不均衡，在创客环境建设、师资培养、内容建设以及课程资源等方面均有待完善和改进。

1.2.2. STEAM 与创客整合课程的具体现状分析

1.2.2.1. STEAM 与创客环境建设差异较大，没有统一标准

由于各地区教育和经济发展不平衡，环境建设的资金投入也有较大的差距，STEAM 与创客环境建设差异较大，根据调查的结果表明,目前部分中小学建设了专用的创客空间或开拓了校外创客基地,不仅有基本的创客设备， 还有物理、化学、生物等学科实验室设备，但部分学校没有创客空间,依托机房进行教学,缺乏跨学科教学的条件，不利于STEAM 与创客整合课程的实践。除此之外，STEAM 与创客设备更新过快且费用较高，各学校无法跟上设备与技术的更新速度。

1.2.2.2. 教师培训的形式与内容比较单一

根据调查数据显示教师对参加 STEAM 与创客教育培训积极性较高，但调查显示教师培训的形式、内容单一且机会较少。在访谈过程中了解到，教师基本没有专门STEAM 与创客教育整合的培训，培训内容主要是创客课程的实践操作，导致教师对于 STEAM 教育理念理解上不到位，教学与评价大部分以实际操作为主。除此之外，STEAM 与创客教师多为由单一学科教师转换的兼任教师，缺少技术能力和多学科知识背景。因而在专业知识与技能方面，要结合 STEAM 教育理念，强化教师们的综合能力、指导学生开展活动的专项能力。

1.2.2.3. 教学内容没有统一标准，教材选用自由度较大

所调查区域的 STEAM 与创客课程的教学内容丰富多样没有统一标准，各区域都有自己的特色课程，如动漫课程是内蒙古西部某市的区域特色课程。创客教材选用的自由度也较大，部分教师为了更好的开展教学活动会自主研发相应的教材，但整体上仍缺乏规范统一的教材,不 利于教师和学校相互参考与学习。教学内容则主要依附 于教师的个人经验，影响创客教育常态化发展和学生的 个性化发展。

1.2.2.4. STEAM 与创客整合课程优质资源较少

目前，STEAM 课程和创客课程的教学资源的数量较多， 但STEAM 与创客课程整合的教学资源是较少的，且质量不高。学校在缺乏优质的 STEAM 与创客教学资源的环境下开展创客课程，很大程度的增加了教师们的负担。 并且，缺乏优质的教学资源不利于学生个性化发展和创新能力的提高。

2.策略与建议

2.1. 通过制定发展规划推进区域 STEAM 与创客教育优质均衡发展

结合该地区 STEAM 与创客教育的发展现状制定符合该区域的发展规划。区域发展规划不仅要着眼于目前STEAM与创客教育的开展现状，更要立足于 STEAM 与

创客教育未来的发展趋势。在制定发展规划时需要格外注意教师的实际需要、基本能力，一线教师最了解学生的学习需求，能为 STEAM 与创客教育发展提出更加切实的建议。区域发展规划在实施的过程应结合实际情况不断调整与修改，切忌照搬照抄。

2.2. 规范学校 STEAM 与创客环境建设

正如谢作如教授提出的创客空间不一定要大，合适即可[3]。创客空间的设备资源也不能盲目的购买，投入的资金应合理的配置设备资源。各学校还可以根据自己的实际情况对原有的机房和实验室等进行改造， 扩充 STEAM 与创客教室的容量。充分利用创客教育的开放性，积极融合 STEAM 教育中的跨学科整合知识，在条件允许的情况下可以带领学生去科技馆或图书馆参观 或开展相应的创客活动。除此之外，各学校可以参考雒 亮、祝智庭的创客空间 2.0 构建模型构建线上+线下创客空间。

2.3. 推荐、编写 STEAM 与创客优秀教材

推荐、编写优秀的教材有利于教师的交流讨论以及教学内容体系的建设。各地区应鼓励各个学校和教师积极的编写适合学校特色的 STEAM 与创客教材，并对优秀教材进行搜集、筛选与评估，将好的教材广泛的推荐、运 用。编写教材时不仅要立足于创客课程的实践操作，更要重视 STEAM 教育的理念渗透，将科学知识和地区文化与创客教育融合。

2.4. 丰富培训内容与方式促进教师专业发展。

STEAM 和创客整合的课程是跨学科探究式的创客教育，培训内容应理论与实践并重，贯彻 STEAM 教育理念，讲解适用于 STEAM 与创客整合课程的教学模式，将培训内容落到实处。培训不仅可以通过传统的讲授方法方式，还可以通过专题培训、网络教研、课题研究和活动比赛等方式以讲授为辅，自主学习为主，培养复合型 STEAM 与创客教师。通过培训在潜移默化中唤起教师内心对 STEAM 与创客教育真正热爱，引导教师们组建专业的 STEAM 与创客团队，在相互交流与分享中促进 STEAM 与创客教育的发展。

2.5. 构建 STEAM 与创客教育的本地教学资源库

以基于 STEAM 教育理念的跨学科资源和创客资源为基础，结合区域的特色资源，加强各地区、各学校、企业和高校之间的合作，建设综合性的本地教学资源库，实现网络资源的共建共享，为学生的个性化学习，教师的课程开展提供高质量的教学资源，有效提升 STEAM 与创客教育的质量。本地资源库的有效使用还需对其进行管理和维护，明确资源成果的产权保护问题，教学资源开发和更新标准，建立健全保护政策，保障本地教学资

源库的建设。

3. 总结

STEAM 与创客整合的课程目前正处于发展探索阶段，通过对内蒙古自治区某市的调查研究了解中国西部地 区 STEAM 与创客教育的发展现状，并提出五项发展策略。但受个人能力以及研究水平所限，本研究还有些不足之处，首先是进行实地走访调查的地区较少，主要采用访谈法进行调查研究，其次是访谈对象都是教研员和教师，缺少学生角度的相关研究，所以，研究结论相对片面，在今后的研究中争取扩大研究对象与研究范围， 为促进中国西部地区 STEAM 与创客教育的发展提供更有针对性的策略和建议。

*内蒙古师范大学一流课程建设项目“信息技术教学设 计”阶段性成果

4. 参考文献

殷朝晖,王鑫. 美国K-12 阶段STEM 教育对我国中小学创客教育的启示[J]. 中国电化教育,2017,(02):42-46+81.

傅骞,王辞晓.当创客遇上 STEAM 教育[J].现代教育技术,2014(10):37-42.

谢作如,刘正云. 做一个灵小可复制的校园创客空间——以温州中学为例[J]. 教育与装备研究,2016,32(03):43-46.

雒亮,祝智庭. 创客空间 2.0:基于 O2O 架构的设计研究[J]. 开放教育研究,2015,21(04):35-43.

王梦珂. 促进核心素养发展的 STEAM 与创客整合课程开发与实践研究[D].华中师范大学,2020.

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

Research on the Current Situation and Strategies of STEAM and Maker Integrated Curriculum in Western China

Yue JIA[1], Mei CHEN[2*]
[1,2]College of education, Inner Mongolia Normal University, Hohhot, Inner Mongolia
Jia Yue@1005574467@qq.com, Chen Mei@ nmchenmei@qq.com

ABSTRACT

Under the background of implementing innovation driven development strategy in China, STEAM education and maker education are both effective ways to cultivate innovative talents. Combining STEAM education with maker education and developing maker education from the perspective of STEAM can better cultivate students' Computational Thinking and innovative practice ability. Through unstructured interviews, field research and classroom observation, this study collects data and materials to grasp the development status of STEAM and maker curriculum in Western China. Through the analysis, it finds out the existing problems and challenges, and puts forward five strategies and suggestions to try to solve the problems and promote the development of STEAM and maker education in Western China.

KEYWORDS

Western China; STEAM education; Maker curriculum; Current situation research

STEM Pedagogies and Curriculum

Looi, C.K., Wadhwa, B., Dagiené, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

初中生 STEM 学习观念调查研究

马媛媛 [1]，周颖 [2*]，朱丹琪[3]
[1] 北京一土学校，中国
[1,2,3] 北京师范大学教育学部，中国
461290383@qq.com，yzh@bnu.edu.cn，595974136@ qq.com

摘要

STEM 教育培养创新型人才，进而提高国家综合实力，保证国家在全球的竞争力。但我国 STEM 教育发展状况参差不齐，处于职业选择关键期的初中生 STEM 学习兴趣较低，促进青少年 STEM 学习，培养 STEM 优秀人才第一步就是了解学生们怎样看待 STEM 学习，即了解他们的 STEM 学习观念。研究通过现象图示学方法调查两所学校共 29 名同学，得出初中生 STEM 学习观念，即学生认为 STEM 学习是：「解决工程问题」、「跨学科综合应用」、「可迁移理解」、「通过集体努力达到自我实现」、「以实践为导向的个人 发展」。同时研究显示，不同整合 STEM 课程类型下，初中生 STEM 学习观念差异较大。

关键词

初中生；整合 STEM 教育；STEM 学习观念

1. 绪论

工业 4.0 时代，引起了全球范围内新一轮技术革新和升级，未来产业对人才需求变化迅猛。全球各国积极改革人才结构，通过教育途径寻求国家新发展，成为提 高国家综合实力的新趋势。培养科技创新人才，已经 成为维持全球经济领先地位的直接动力，应从提升国 家竞争力角度，看待人才培养（ Mohr-Schroeder, Cavalcanti, & Blyman, 2015）。STEM 教育着眼于培养创新型复合人才，是未来教育的发展方向（金慧和胡盈滢，2017）。

尽管我国 STEM 教育在国家政策全力支持下发展迅速，但有调查显示（赵姗姗，2005；薛品和赵延东，2015），我国许多处在职业选择关键期的中学生，不愿在 STEM 领域开展相关学习和工作，我国严重缺乏 STEM 领域专业人才。STEM 教育承担着越来越多人才培养的责任，为此，在中学这个职业选择关键期，促进青少年学习 STEM，使他们接受优质 STEM 教育，满足国家发展需求，帮助他们成为 STEM 文化公民，加入 STEM 劳动力迫在眉睫（Dugger, 2010）。而激发学生学习，培养 STEM 优秀人才第一步就是了解学生们怎样看待 STEM 学习，即了解他们的 STEM 学习观念。

2. 文献综述

2.1. *STEM 教育*

STEM 是科学（Science）、技术（Technology）、工程（Engineering）和数学（Mathematics）四门学科的简称（刘阳、王志博和王会丽，2019）。本研究选取整 合角度阐述 STEM 教育定义与课程类型。「整合性STEM」教育是指，通过整合与技术相关的工程实践和工程设计，实践科学或数学学科内容的教学（Schnittka, 2017）。整合 STEM 教育实践，依靠课程实现学习目标，依据对科学、技术、数学、工程四门课程融合程 度的不同，分为「相关课程」、「融合课程」、「核 心课程」（吕延会，2017）。教师可在不同教学情境 下选择不同类型的课程进行实践，学习者在不同整合 性课程下所获得的知识与能力是否不同有待研究者进 行更深入探讨。

国内外 STEM 研究已经涉及教师拥有的 STEM 观念（Kloser et al., 2018; Radloff & Guzey, 2016; Ring et al., 2017; Sevil & Aslan-Tutak, 2016），及学生对 STEM 学习经历的认知（Mullet, Kettler, & Sabatini, 2018），已触及与学生学习经历相关的专业问题，但研究目前仅 局限于分析教师与学生怎样看待 STEM 和对 STEM 教育本质的理解，并未对学生STEM 学习看法，即观念的深层次挖掘。有研究显示（Mullet, Kettler, & Sabatini, 2018），教师或学生对 STEM 学习的深层次理解，即

「STEM 学习观念」，或许会对 STEM 学习效果有一定的影响，且国内 STEM 研究较重视 STEM 课程开发和关注 STEM 课程的学习效率（王涛、马勇军和王晶莹，2018），而本文着眼于深入探究中学生对STEM 学习本身的理解，即中学生的 STEM 学习观念

。

2.2. 学习观念

学习观念是指学习者对学习的认知或解释（Richardson, 1999），对学习经历的看法（Tsai, 2004）。STEM 学习观念是学习者对 STEM 学习的认知与解释， 对STEM 学习经历的看法。拥有较低层级学习观念的学习者认为，学习是刻板式记忆、做题，拥有较高层级学 习观念的学生认为学习是对知识与学习方法的深层次 理解。

研究发现，学习观念与学习方法（ Chiou & Liang, 2012）、学习自我效能感（Liang, 2015）、以及学术成就（Peterson, Brown, & Irving, 2010）有关，学习领域不同，学生拥有的学习观念也不同（Gunilla, 1998; Sadi,
2015; Tsai, 2004, 2009; Tsai & Kuo, 2008）。大多数学生拥有的理科课程学习观念层级较低（Tsai, 2004）， 同时关于整合的理工科课程，学生的学习观念研究较 少。从这个观点来看，STEM 学习观念研究，既能为学习观念不同领域议题提供新思路，同时又能为教育者和研究者提升理工科类学生学习观念带来新想法，让他们能够更好地理解学生的学习经验，从侧面了解教 师教学成果与课程实施效果。教师还可以进一步在此 基础上，调整课程结构以改善学习环境，提高教学效 率，增加学生对理工科科目的学习兴趣，让他们的学 习更上一层楼（McInnes et al., 2010）。

3. 研究设计

3.1. 研究内容

初中生处于人生成长关键转折期，其认知发展会通过 学习有较大的变化（ 王晨菡、谭积斌和曾卉玢， 2018）。应加深对初中生 STEM 学习观念的深入探讨，填补国内关于此类研究的空白，推进 STEM 教育相关研究的步伐。本文通过现象图示学研究方法，深
入分析初中生 STEM 学习观念，探讨不同整合 STEM 教育课程类别下初中生 STEM 学习观念的差异，理解学生所持 STEM 学习观念本质，明确初中生在整合 STEM 学习环境中学习观念在性质上的异同。

3.2. 研究方法

现象图示学是旨在了解人们对身边现象的解释及看法 的定性研究研究方法（Martoňák & Tosatti, 1994），其目的为描述人们周围各种现象的本质（Marton, 1981）。现象图示学与学生学习联系紧密，大多数学习 观念 研究 （ Gunilla, 1998; Marton, 1981; Marton,Dall'Alba, & Beaty, 1993）均采用现象图示学方法开展研究。Marton 等人（1981）提出，现象图示学研究步骤如下：（1）确定样本，数量通常在 10-30 即可；
（2）确定数据收集方法—最常见的为半结构化访谈或小组座谈；（3）将被访谈者语音转录为文字，对转录文本进行文本分析；（4）确定概念的「描述类别」 或重要的变化实例；（5）确定「变化的维度」 与类别之间的关系；（6）将类别和维度表示为「结果空间」。

3.3. 数据收集

专业人员对每一位研究样本学生，通过半结构化访谈 方式，尽量从客观角度出发，避免带入主观情绪，防 止访谈者效应，单独采访收集数据。访谈问题尽量从 不同角度了解受访者对于某一事物的看法与见解，从 数据收集角度保证了研究的信、效度。指导性访谈问 题从 Marshall（1999）和 Tsai（2004）的研究中修改而来，如下：

（1） 你认为什么是 STEM 学习？

（2） 你是怎么学习 STEM 课程的？

（3） 你怎么知道你在学习 STEM 课程？

（4） 在 STEM 课程中，你学习到了什么？

（5） 你对 STEM 学习的感觉是什么？

这五个问题分别从学生怎样看待 STEM 学习，学生的 STEM 学习方式，学生印象较深的 STEM 学习情景，以及学生在 STEM 课程中学习到的内容，对学生的 STEM 学习进行了调查。

3.4. 数据分析

本研究在数据分析中的类别提取依据了下述两类研究 结论。其一，根据下一代科学标准（NGSS），和部分学者阐述的 STEM 学习过程特征（吕延会，2017）总结得出的 STEM 学习过程：调研与确定工程问题；设计修正；小组合作，发挥创意，动手完成设计，产出

过程性或总结性作品；反思分享。其二，已有的学习 观念相关研究结果（ Asikainen, Virtanen, & Parpala, 2013; Gunilla, 1998; Täks, Tynjälä, & Kukemelk, 2016;
Tsai, 2004, 2009），如科学学习观念包含七类：记忆、做题、考试、增加知识、理解、应用、和应新的方法看，前三类为低阶观念，后四类为高阶观念（ Tsai, 2004）。

在数据分析过程中利用研究者三角交叉方式，两位专 业人员分别对文字信息分析，对比总结学生在访谈中 对 STEM 学习观念回答的异同， 分析学生拥有的STEM 学习观念类别。

3.5. 研究对象

研究分别从 A、B 学校，共选取 29 名七年级同学作为调查样本，包括 A 校七年级学生 15 名（男生 9 名，女生 6 名）和 B 校七年级学生 14 名（男生 6 名，女生 8名）。

A、B 两所学校均在 2019-2020 秋季学期，为七年级学生开设 STEM 课程，A 校学生通过体验完整 STEM 学习过程，完成「生物科考站」项目式学习任务，学习解决问题，为核心课程。B 校学生动手实践，拍摄、制作创意视频，完成坦克与飞机模型制作，并为体验完 整 STEM 学习课程，为融合课程。

4. 研究结果

分析发现，A 学校学生 STEM 学习观念为以下五个类别： 学习是「解决工程问题」、「跨学科综合应用」、「可迁移理解」、「通过集体努力达到自我实现」、「以实践为导向的个人发展」。

B 学校学生 STEM 学习观念为以下六个类别：学习是「解决问题」、 「知识运用」、「技术运用」、「深度理解」、「通过集体努力达到自我实现」、「以实 践为导向的个人发展」。

整体分析总结得出：「解决工程问题」相较「解决问 题」更能贴合整合 STEM 教育中学习过程描述，「可迁移理解」含义包含「深度理解」，「跨学科综合应 用」含义包含「知识运用」与「技术运用」。故初中 生 STEM 学习观念的研究结果为： 「解决工程问题」、「跨学科综合应用」、「可迁移理解」、「通 过集体努力达到自我实现」、「以实践为导向的个人发展」共 5 类。这 5 个 STEM 学习观念相互关联， 体现了整合 STEM 教育的目标，表明 STEM 教学可以让学生在解决工程问题过程中实现较高层级的有意义学习。

5. 讨论

整合 STEM 课程作为 21 世纪国家素质教育重要举措之一，通过跨学科整合知识、方法、技能，鼓励学生基于真实世界的问题开展学习。

A 学校 STEM 课程设置符合核心课程特征，较重视工程实践过程的完整性，以项目式学习为依托，让学生 在合作的实践过程中提高综合素养，学会利用跨学科 知识、方法、技术解决与现实生活相关的问题。B 学校课程设置为融合课程，较重视学生的动手实践过程， 工程实践部分较少。在不同课程设置下，学生表达了 不同程度的 STEM 学习观念，说明调查学生的 STEM 学习观念有可能反映出 STEM 课程的差异。

对比理工科类学习观念与 STEM 学习观念，初中生 STEM 学习观念的 5 个类别均处于较高层级。在 STEM 学习过程中，学生认为自己的理科学习不再是刻板式 记忆、做题，而是在动手实践过程中，达到对知识与 学习方法的深层次理解， 学会在综合运用知识、方法、技术解决问题的同时，提高自己各方面的能力， 而且较少学生会表达对学习的消极情绪。也就是说， STEM 学习能够促使学生将知识，迁移应用到新的情景，让他们产生内在动机激励自己的学习过程，体会 到知识学习与应用的乐趣。

整合 STEM 教育将科学与数学内容应用于工程与技术实践，这样的课程设置为理工科类知识提供了新的学 习情境，学生不再拘泥于简单的概念输入，而是在真 实的问题解决环境中，达到知识更深层次的理解与运 用。

基于本文研究结果，建议教师在课程设计层面，保证 课程融合的深度，秉持实践大于理论的原则，不仅重 视 STEM 课程中的动手实践，更要将课程定位在更深层次的解决问题环节，重视工程设计，确保学生体验 完整的 STEM 学习环节，不将教学流于形式。其次， 教师应在教学中，真正深入了解学情，设置合理的真 实学习情境，使学生与之产生连接，让学生在真实且

与自己息息相关的背景下，通过小组合作的方式，深 层次提升学生综合能力。

STEM 课程以其独特的学科特征，也可以为传统的理科学习，提供一些新的教学灵感，为理工科类中小学教 育改革提供新思路。建议教师与研究者考虑将 STEM 学习作为改善理科类学生学习观念的手段之一，开发 融合程度较高的整合 STEM 项目，将科学知识合理且巧妙地设计为解决实际问题必不可少的基础，让他们 在实际情境下，应用知识，自然会提高对知识的理解 程度， 改变死记硬背的习惯，进而达到更高层次的学 习境界

。

6. 展望

本研究样本来源较单一化，未对学习观念表达缺失原 因进行分析，同时并未深入探究为什么学生会拥有较 高层级的学习观念，并未了解影响 STEM 学习观念的因素。希望未来能够增加样本数量，将中学生 STEM 学习观念研究从质性分析迈向量化分析阶段，深入探 讨 STEM 学习给学生带来的改变。

7. 参考文献

王秀民（2018）。量化 **vs** 质性研究分析。2018年8月19 日，取自 https://wenku.baidu.com/view/18256104ff4733687e21af45 b307e87101f6f8d7.html。

王涛、马勇军和王晶莹（2018）。我国STEM教育现状研究——基于2011-2017年核心期刊文献的分析。世界 教育信息，**31**，23-28。

王晨菡，谭积斌和曾卉玢（2018）。基于内容分析法的STEM教育研究综述。浙江教育科学， 10-15。

吕延会（2017）。STEM教育的核心精神。当代教育科 学， 16-19。

刘阳、王志博和王会丽（2019）。中小学STEM教育的发展现状及启示。中小学电教:教师版， 3-6。

金慧和胡盈滢（2017）。以STEM教育创新引领教育未来——美国《STEM2026：STEM教育创新愿景》报告的解读与启示。远程教育杂志，**35**，17-25。

赵姗姗（2005）。浅析影响当前青少年职业理想的因素。邢台学院学报，**20**， 37-40。

薛品和赵延东（2015）。青少年职业期望及影响因素研究。教育学术月刊，**11**，76-83。

Asikainen, H., Virtanen, V., & Parpala, A. (2013). Understanding the variation in bioscience students' conceptions of learning in the 21st century. *International Journal of Educational Research*, 62, 36-42.

Chiou, G. L., & Liang, J. C. (2012). Exploring the Structure of Science Self-Efficacy: A Model Built on High School Students' Conceptions of Learning and Approaches to Learning in Science. *The Asia-Pacific Education Researcher*, 21(1), 83-91.

Dugger, W. E. (2010). *Evolution of STEM in the United States*. Paper presented at the 6th Biennial international conference on technology education research, Gold Coast, Queensland, Australia.

Gunilla, E. (1998). Students' Conceptions of Learning in Different Educational Contexts. *Higher Education*, *3*(35), 299-316.

Kloser, M., Wilsey, M., Twohy, K., Immonen, A., & Navotas, A. (2018). "We Do STEM": Unsettled Conceptions of STEM Education in Middle School S.T.E.M. Classrooms. *School Science and Mathematics*, *118*(8), 335-347.

Liang, J. C. (2015). Conceptions of Memorizing and Understanding in Learning, and Self-Efficacy Held by University Biology Majors. *International Journal of Science Education*, *37*(3), 446-468.

Marshall, D., & Woolnough, S. (1999). Students' Conceptions of Learning in an Engineering Context. *Higher Education*, *3*(38), 291-309.

Marton, F. (1981). Studying conceptions of reality a metatheoretical note1. *Scandinavian Journal of Educational Research*, 25(4), 159-169.

Marton, F., Dall'Alba, G., & Beaty, E. (1993). Conceptions of learning. *International Journal of Educational Research*, (19), 277-299.

Martoňák, R., & Tosatti, E. (1994). Path-integral Monte Carlo study of a model two-dimensional quantum paraelectric. *Physical Review B Condensed Matter*, 49(18), 12596-12613.

McInnes, K., Howard, J., Miles, G., & Crowley, K. (2010). Differences in Adult-Child Interactions During Playful and Formal Practice Conditions: An Initial Investigation. *Psychology of Education Review*, *34*(1), 14-20.

Mohr-Schroeder, M. J., Cavalcanti, M., & Blyman, K. (2015). *A Practice-Based Model of STEM Teaching*.

Mullet, D., Kettler, T., & Sabatini, A. (2018). Gifted Students' Conceptions of their High School STEM Education. *Journal for the Education of the Gifted*, *1*(40), 60-92.

Peterson, E., Brown, G., & Irving, S. (2010). Secondary School Students' Conceptions of Learning and their Relationship to Achievement. *Learning & Individual Differences*, *20*(3), 167-176.

Radloff, J., & Guzey, S. (2016). Investigating Preservice STEM Teacher Conceptions of STEM Education.

Journal of Science Education & Technology, *25*(5), 759-774.

Richardson, J. (1999). The Concepts and Methods of Phenomenographic Research. *Review of Educational Research*, *69*(1), 53-82.

Ring, E., Dare, E., Crotty, E., & Roehrig, G. (2017). The Evolution of Teacher Conceptions of STEM Education Throughout an Intensive Professional Development Experience. *Journal of Science Teacher Education*, *28*(5), 444-467.

Sadi, O. (2015). The Analysis of High School Students' Conceptions of Learning in Different Domains. *International Journal of Environmental & Science Education*, *10*(6), 813-826.

Schnittka, C. (2017). STEM Road Map: A Framework for Integrated STEM Education. *Journal of Educational Research*, *110*(3), 317.

Sevil, A., & Aslan-Tutak, F. (2016). STEM Images Revealing STEM Conceptions of Pre-Service Chemistry and Mathematics Teachers. *International Journal of Education in Mathematics Science & Technology*, *4*(1), 56-71.

Täks, M., Tynjälä, P., & Kukemelk, H. (2016). Engineering students' conceptions of entrepreneurial learning as part of their education. *European Journal of Engineering Education*, 1(41), 53-69.

Tsai, C. C., & Kuo, P. C. (2008). Cram School Students' Conceptions of Learning and Learning Science in Taiwan. *International Journal of Science Education*, *3*(30), 353-375.

Tsai, C. C. (2004). Conceptions of Learning Science Among High School Students in Taiwan: A Phenomenographic Analysis. *International Journal of Science Education*, *26*(14), 1733-1750.

Tsai, C. C. (2009). Conceptions of Learning Versus Conceptions of Web-Based Learning: The Differences Revealed by College students' conceptions of context-aware ubiquitous learning: A phenomenographic analysis College Students. *Computers & Education*, *53*(4), 1092-1103.

Looi, C.K., Wadhwa, B., Dagienė, V., Liew, B.K., Seow, P., Kee, Y.H., Wu, L.K., & Leong, H.W. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Teachers Forum 2021.* Singapore: National Institute of Education.

An Investigation on STEM Learning Conceptions of Junior School Students

Yuan-yuan MA[1], Ying ZHOU[2*], Dan-qi ZHU[3]
[1]Beijing ETU School, China
[1,2,3]Faculty of Education, Beijing Normal University, China
461290383@qq.com, yzh@bnu.edu.cn, 595974136@qq.com

ABSTRACT

STEM education cultivates innovative talents to improve the country's overall strength and ensure the country's global competitiveness. However, the development of STEM education in China is uneven. Junior high school students who are in the critical period of career choice have low interest in STEM learning. The first step to promote STEM learning among teenagers and cultivate outstanding STEM talents is to understand how students view STEM learning, namely, to understand their concept of STEM learning. This survey of 29 students in two schools by phenomenological graphology shows that the STEM learning concepts of junior high school students are as follows: "solving engineering problems", "interdisciplinary comprehensive application", "transferable understanding", "achieving self-realization through collective efforts" and "practice-oriented personal development". Meanwhile, the research shows that STEM learning concept of junior high school students varies greatly under different integrated STEM curriculum types.

KEYWORDS

Junior high school students, Integrating STEM Education, conception of STEM learning

CTE-STEM 2021

5TH APSCE INTERNATIONAL CONFERENCE ON COMPUTATIONAL THINKING AND STEM EDUCATION

https://cte-stem2021.nie.edu.sg/
cte.stem2021@nie.edu.sg