國立成功大學

工程科學系

博士論文

Department of Engineering Science
National Cheng Kung University

互動性平台對於大學生程式設計的影響

Interactive e-Learning Platform's Impact on the Programming Skill to University Students

研究生：黃聖博　　Student: Sheng-Bo Huang

指導教授：賴槿峰　　Advisor: Chin-Feng Lai

中華民國 112 年 7 月

國立成功大學

博士論文

互動性平台對於大學生程式設計的影響

Interactive e-Learning Platform's Impact on the Programming Skill to University Students

研究生：黃聖博

本論文業經審查及口試合格特此證明

論文考試委員：

指導教授：

單位主管：

（單位主管是否簽章授權由各院、系（所、學位學程）自訂）

中 華 民 國 112 年 7 月 28 日

Interactive e-Learning Platform's Impact on the Programming Skill to University Students

Huang, Sheng-Bo

A Dissertation submitted to the Faculty of the
Department of Engineering Science
In partial fulfillment of the requirements for the Degree of
Doctor of Philosophy
College of Engineering
National Cheng Kung University Tainan, Taiwan
July, 2023

Approved by:

Wei-Zhong Wang
Chi-Cheng Chang
Yu-Lin Jeng
Chen, Hong-Ren

Dissertation Advisor:

Department Chairman:

Wen-Fung Pan

摘要

現今數位化的時代，資訊和通訊技術（ICT）的概念已經融入人們的日常生活之中，這也顯示具有資訊和通訊技術相關的技能，對於大學生以及未來的職業生涯有著高度的關聯，因此程式設計是他們大學期間最重要的課程，程式設計課程不僅培養學生電腦科學的技能，而且還能夠培養學生的運算思維（CT）、解決問題能力和批判性思維，這些能力對於現今數位化時代是不可缺少的。然而電腦科學是培養學生運算思維能力的一種好方法，但教學生學習程式設計並不是一件容易的事，因為程式設計是一種專業能力，也需要一定的基礎知識，這也顯示出程式設計對初學者學習具有一定的難度，因此，學習者在練習程式設計時，很容易會遇到很多錯誤訊息，而這也會影響學生的學習動機，以至於降低他們的學習參與度。

然而，為了提高學生的 ICT 素養，本研究提出了兩個互動式學習平台和學習活動設計，以探討學生的學習成績和參與度的變化。1）即時程式設計教學輔助系統，該系統整合了即時回饋模組、異常檢測模組、使用者收集模組和過濾模組，因此本系統會持續監測學生的錯誤資訊，並在學生遇到錯誤訊息時提供一些建議，以讓學生作為學習上的參考。2）做筆記的互動式學習平台，在本實驗中，此系統具有資源分享和社交模組等功能，以刺激學生的互動，此外，為了讓學生有更多與同儕交流的機會，系統也將學習活動作為教學設計的一部分，因此教學設計以同儕學習理論做為基礎，並將教學設計與同儕學習理論相結合，形成本實驗的實驗設計，除此之外，在課程中也設計了一系列的學習活動，以觀察本實驗中學生的學習動機和學習成效的變化。整體而言，二個互動式學習平台中都將整合不同的教學策略並融入到課程中，收集資料來分析二個互動式學習平台對學生的影響。綜上所述，二個互動式學習平台的設計對學生的學習成效產生了正向的顯著的差異。另一方面，結果顯示也指出二個互動式學習平台對學習參與度都有正向的影響，但沒有達到顯著差異。

關鍵字: 資訊素養、計算機思維、互動式學習平台、學習動機

ABSTRACT

In the digital age, Information and Communication Technology (ICT) is used in every aspect of people's daily lives. ICT-related skills are critical for university students to succeed in their academic lives and future careers. Software programming is one of the most important trainings during their university studies. It not only teaches students computer science skills but also develops students' computational thinking (CT), problem-solving, and critical thinking. These capabilities are beneficial to a better life in this digital time. Although computer science is an effective tool to develop students' computational thinking skills, teaching students to learn software programming is not an easy task. It involves a high level of expertise and requires a certain level of basic knowledge. This highlights the challenges of learning software programming for beginners. Learners will encounter many error messages when practicing coding, which can make students feel unmotivated and reduce their learning engagement.

To improve students' ICT literacy, this study proposes two interactive e-learning platforms and learning activity designs to explore the changes in students' learning performance and engagement. (1) A real-time programming teaching support system: This system integrates a real-time response module, an exception detection module, a user collection module, and a filter module. Therefore, the system will continuously monitor students' error messages and provide suggestions to students as a reference when they encounter them. (2) A note-taking interactive e-learning platform. In this experiment, the system displays resource sharing and social modules to stimulate student interaction. To provide students with more opportunities to communicate with their peers, the system includes learning activities as part of the instructional design. The instructional design integrates instructional design and peer learning theory to form the experimental design of this experiment. Moreover, a series of learning activities were arranged in the curriculum to observe the changes in students' motivation and academic performance in this experiment. Each interactive e-learning platform incorporates different instructional strategies into the curriculum. Data were collected to analyze the impact of each interactive e-learning platform on students. The design of the two interactive e-learning platforms had a positive and significantly different impact on students' learning performance. On the other hand, the results indicate that each interactive e-learning platform positively impacted learning engagement but did not reach a significant difference.

Keyword: ICT literacy, Computational thinking, Interactive e-learning platform, Learning motivation

Acknowledgements

經過了漫長的博士生活，終於完成博士論文以及口試，在這段期間經歷不少的事情。首先要感謝實驗室的每一位同學與老師，為我的研究上帶來許多的幫助與想法，在我就讀的期間也為我的生活帶來許多的樂趣。另外，我也要特別感謝鄭鈺霖老師，從我一開始什麼都不會的階段，一直不斷的給我建議與細心的教導，過程中也花費老師非常多的時間，但老師還是很有耐心的教學，我也從過程中學習到一些相關的專業技能，到現在能夠獨立作業，真的非常感謝老師這一路上的幫忙與指導。

另外，在就讀博士這一段期間，我也去了千里馬計畫，很感謝譚老師在加拿大一路的上的幫忙，讓我在加拿大遊學的期間也有回到家裡的感覺，也讓我的國外生活安心了許多，加拿大遊學這段期間也成為我人生最難忘的回憶。

此外，我也要感謝各位口試委員的幫忙，透過口試委員的專業意見讓我的博士論文能夠更加的完善，也非常感謝口試委員們指出論文的問題，讓我的論文品質也能夠更進一步的提升。

最後非常感謝我的女朋友與家人，不論發生什麼事，他們總是會給予我鼓勵與支持，讓我可以有動力的完成我的學位，特別是我在國外生活這段期間，不斷的給我鼓勵與幫助，這些鼓勵讓我更加有勇氣面對國外生活的各種挑戰。

黃聖博 謹誌

中華民國 112 年 7 月

Table of Contents

Chapter 1. Introduction

1.1 Background and Motivation

In the current digital age, Information and Communication Technology (ICT) has become a necessity. Most people use ICT tools in their daily lives, such as mobile phones, computers, tablets, etc. To develop ICT-related skills among citizens, the Taiwanese government has actively promoted science-related policies. These include encouraging the integration of information technology in school curriculums and encouraging schools to establish technology courses in computer science, artificial intelligence (AI), and the Internet of Things (IoT) (Chu et al., 2020; Yuan, 2021). These literatures highlight the importance of ICT-related skills (Chu et al., 2020; Yuan, 2021), and therefore, ICT-related subjects have become popular university courses. Therefore, computer science education has become an important issue. The main reason is that computer science can develop various thinking skills in students, such as innovative thinking, project management, the ability to solve new problems, team building, communication skills, etc. (Giannakos et al., 2017; Harlow et al., 2018). This suggests that developing students' computer science skills helps enhance students' cognitive skills and their ability to communicate with their peers. As a result, the field of computer science has expanded, where computational thinking was proposed. The central concept of computational thinking is that humans can organize information and solve problems strategically and effectively through computer science (Dougherty, 2012; Google, 2015; Wing, 2006). Based on these discussions, ICT-related subjects can be considered key disciplines in the digital age. ICT derives from many different disciplines, such as computer science, computational thinking, etc. These disciplines not only develop students' information technology skills but also train problem-solving skills and apply them to daily life.

ICT-related developments have become important worldwide, with many countries actively promoting ICT-related curricula and government funding a large budget to bring ICT into schools. Fraillon et al. (2018) found that the Korean government is revising the national curriculum to respond to the needs of society. They further found that the Ministry of Education provides the curriculum framework for schools to improve ICT literacy for students to respond to the changing needs of the country (Fraillon et al., 2018). Similar findings have been elicited through other research. In the Korean government's view, ICT literacy is the basic ability to effectively solve the issues of the digital age; this comprises understanding the principles of ICT and developing the knowledge needed by ICT users, such as computer science and programming (kim et al., 2019). Meanwhile, computational thinking has become a core skill that students need, primarily due to the rapid emergence of technologies with artificial intelligence and the Internet of Things (Kim et al., 2019; Wing, 2006). In Germany, the government started funding schools with federal funds beginning in 2019 with a budget of €5 billion to integrate ICT into schools with established models for primary and secondary school students (Fraillon et al., 2018). According to Barr and Stephenson (2011), Grete and Yadav (2016) and Kim et al. (2021), the trend in ICT literacy education is to develop computational thinking skills while emphasizing that ICT literacy is related to computational thinking skills. This is where computer science knowledge has been extended to include computational thinking concepts. Recently, ICT literacy has been increasingly emphasized in teaching computer science, most often using ICT tools to solve problems in all disciplines; this also emphasizes that computer science principles are essential skills to daily life (Barr & Stephenson, 2011; Yuliana et al., 2021).

Based on the above discussion, many countries actively promote ICT literacy. The main reason is the ability to develop the knowledge students need, such as problem-solving skills, computer science, and computational thinking, which are even more critical in the digital age. However, even though ICT literacy is necessary, the development of students learning ICT literacy is challenging. Teachers require a certain level of background or the redesign of their curriculum to integrate ICT tools into the curriculum. Gao (2021) and Hayitov (2020) found that teachers play a crucial role because they are actively involved in all stages but have less influence on the ICT integrated curriculum. If teachers have insufficient knowledge of computer science and information technology, such as using ICT tools, or other tools to organize the curriculum, this causes challenges in effectively developing ICT student literacy. Furthermore, programming science is a way to develop computer science, and students require highly intensive practical programming and understanding of basic knowledge and concepts. However, due to limited class time, it is quite difficult for students to fully comprehend all the material in class. (Aleksić & Ivanović, 2016). Lai, Jeng, and Huang (2020) and Kelleher and Pausch (2005) found a similar circumstance: programming is a professional skill that is challenging for beginners to learn because they encounter several different learning issues. These issues can include not recognizing error messages, syntax errors, command errors, etc., which lead students to be unable to understand the course content effectively and to complete the learning tasks independently.

1.2 Purpose of the Study

The above research background and motivation indicate that ICT literacy is receiving attention from various countries while governments are actively promoting ICT education. Moreover, ICT literacy improves students' thinking skills and helps them develop problem-solving skills and computational thinking. Several different teaching strategies have been proposed to develop students' ICT literacy, such as integrating ICT tools into school curriculums and developing computer science skills through programming. However, programming courses and ICT tools require a certain level of background knowledge. Moreover, research indicates that programming is challenging for beginners (Aleksić & Ivanović, 2016; Lai, Jeng & Huang, 2020; Kelleher, Pausch, 2005; Linder, Abbott & Fromberger, 2006; Fletcher & Lu, 2009); it takes much time to understand the concepts and intensive practice, where programming is an abstract subject that can make it challenging for students to effectively understand course content while encountering many learning issues. The present study aims to improve these issues.

(1) As programming is abstract and difficult to understand, it is challenging for students to improve their learning performance.

(2) Students lack motivation to learn programming.

Based on the above discussion, this study aimed to integrate different learning interactive e-learning platforms into the curriculum to improve student learning performance and motivation. Therefore, this study has four research objectives, the details of which will be described in the next section.

1. To explore whether a real-time programming teaching support system can effectively enhance students' programming learning motivation.
2. To explore whether a real-time programming teaching support system can improve students' programming learning performance.
3. To explore whether note-taking interactive e-learning platforms can effectively enhance students' programming learning motivation.
4. To explore whether note-taking interactive e-learning platforms can improve students' programming learning performance.

1.3 Research Objectives and Research Questions

Based on the study's purpose, ICT literacy can enhance students' thinking skills and help develop problem-solving skills and computational thinking. Learning programming is one of the more common ways to achieve this. One of the many teaching strategies to develop students' programming skills is integrating ICT tools into teaching to develop students' programming skills. Despite the relevance of programming to the development of students' ICT literacy, teaching students programming effectively is not an easy task. Research indicates that programming is not easy for beginners; it takes much time to understand the concepts and gain intensive practice. Furthermore, programming is an abstract subject that can make it challenging for students to understand course content while encountering several learning issues (Aleksić & Ivanović, 2016; Lai, Jeng & Huang, 2020; Kelleher, Pausch, 2005; Linder, Abbott & Fromberger, 2006; Fletcher & Lu, 2009). The present study aims to improve these issues. Many factors contribute to the barriers to students' programming learning. Therefore, to enhance students' programming skills, this study proposes a series of teaching strategies and an integrated interactive e-learning platform to help students learn programming. Two experiments were conducted in this study. The first experiment was to provide a real-time programming teaching support system and scaffolding theory to help students learn programming. The second experiment involves a note-taking learning platform and integrated interactive functions that can help students interact with their peers during learning activities while promoting student motivation. This study explored different learning strategies and e-learning platforms that can positively impact students' programming learning. The research questions are presented below:

1 Does the e-learning platform improve student learning performance?

2 Does the e-learning platform improve student learning engagement?

1.4 Research Contribution

The purpose of the study experiments was to support students in achieving better learning performance in programming learning while improving programming learning motivation. Based on the experiments, we explore which approach will best improve students' abilities. Additionally, teachers can refer to this study regarding the learning material and teaching activity design. This can help teachers to more effectively develop student programming skills. Regarding learning motivation, the background and motivation sections indicate that programming is more likely to make students less motivated to learn. Thus, teachers can consider the study results in modifying teaching strategies to improve student motivation.

Chapter 2. Literature Review

2.1 ICT literacy

Information and Communication Technology (ICT) literacy has been recognized as a key 21st-century skill. As an interdisciplinary technical skill, ICT competency is related to thinking skills. The authors suggest that it can be developed in children at an early age (Binkley et al., 2012; García-Peñalvo & Cruz-Benito, 2016). Kim, Ahn, and Kim (2019) and Kim, Kil, and Shin (2014) suggest that ICT can be seen as the most essential skill of the 21st century as more and more people tend to use ICT to gain knowledge and solve problems. It has been defined in several forms, such as collecting and using digital information-based knowledge. Some research notes that ICT literacy is defined as an individual's ability to use computers, create and communicate, including technology and process digital information, while ICT literacy is domain-specific, which means that it can be used in multiple domains (Carretero, Vuorikari, & Punie, 2017; Ferrari, 2013; Scherer & Siddiq, 2019). In addition to the development of the digital age, the European Commission's scientific proposed the European Digital Competence Framework known as DigComp. This new ICT literacy framework proposes five main skills: (1) information and data literacy; (2) communication and collaboration; (3) digital content creation; (4) security; (5) problem-solving (Carretero, Vuorikari, & Punie, 2017; Ferrari, 2013; Scherer & Siddiq, 2019).

The above literature indicates that ICT literacy is an essential skill for everyone. This is primarily because the acquisition of ICT literacy has a positive impact on the ability to use and process digital information, and this skill can be used in many fields. Therefore, ICT literacy can be seen as technical and digital competence. Santos et al. (2019) suggest that ICT literacy is a factor that influences students' school performance. The Internet of Things has been used as a teaching tool in studies in order to develop students' ICT literacy studies, which also indicates that teachers and parental support can have a positive impact on ICT literacy (Santos et al., 2019). Furthermore, some researchers suggest that ICT literacy can help protect children from the negative effects of media and reduce information inequality. This highlights the importance of ICT. These literatures also highlighted that to develop ICT literacy, two factors can facilitate the development of ICT competence: (1) ICT infrastructure and (2) ICT competence of teachers. However, if students do not have ICT literacy, it is challenging to learn better than before without ICT literacy (Lau & Yuen, 2014; Lee, Chae, 2012).

Overall, ICT literacy is a broad concept that encompasses both technical and digital competencies. Many scholars have explored the impact of ICT literacy levels on achievement. Some studies have indicated that ICT literacy is a factor in school performance that affects the acquisition of knowledge and problem-solving; the collection and use of knowledge based on digital information through ICT tools is a common way of doing so (Kim, Ahn & Kim, 2019; Kim, Kil & Shin, 2014; Scherer & Siddiq, 2019; Carretero, Vuorikari, & Punie, 2017; Ferrari, 2013; Santos et al., 2019). Therefore, this study proposes a series of teaching experiments and integrates different e-learning platforms as ICT tools. This study used two e-learning platforms, each with different functions, including real-time support messages, peer note-taking, etc. Students can learn and read learning materials and interact with their peers through the various functions designed to enhance student achievement and motivation through various learning designs.

2.2 Learning Motivation

2.2.1 Intrinsic and Extrinsic Motivation

There are many ways to foster students' motivation to learn, such as learning strategies or learning environments. Puspitarini and Hanif (2019) found that games can be used as a learning strategy to stimulate students' motivation because learning materials contain a lot of interactive media and innovative materials; further, teachers can act as facilitators in using different learning strategies to promote students' external motivation. In the learning environment, motivation can be seen as a factor that affects students' learning performance. This is mainly because when students are not motivated to learn, it reduces their willingness to learn and affects the learning activities of the participants (Aleksić & Ivanović, 2016; Fotaris et al., 2016; Pilkington, 2018). Puspitarini and Hanif (2019) found that motivation has a positive impact on learning and when the learning process does not effectively attract students, it affects their understanding of the learning material. Since this can decrease motivation, the authors suggest that providing different media can attract students' interest in learning. In contrast, learning resources typically use books, which can easily make students feel bored (Puspitarini & Hanif, 2019). This highlights the importance of student motivation because it correlates with academic achievement. However, there are two factors affecting motivation. Ryan and Desi (2020), Shi, Tong, and Long (2021), and Ryan and Deci (2000) suggest that motivation for learning comprises two factors: extrinsic and intrinsic motivation. Extrinsic motivation is defined by external regulations that involve rewards and punishments enforced from outside. Thus, it can be considered a form of non-autonomous motivation. Intrinsic motivation is defined as one's own intrinsic interests and enjoyment. Therefore, researchers suggest that intellectual stimulation facilitates higher levels of intrinsic motivation. This means that teaching methods are related to the impact on intrinsic and extrinsic motivation and that the learning environment should encompass strategic pedagogy and technological capabilities to promote motivation (Bolkan, 2015; Kirschner et al., 2004). The authors used Augmented Reality (AR) as a teaching strategy after an experiment; the results showed increased intrinsic motivation. Further exploring each dimension showed significant differences in the motivation of attention, satisfaction, and confidence factors (Khan et al., 2019). In another research, interaction with peers was found to be a strategy for promoting student motivation. Smith et al. (2021) found that interaction with peers has been a factor in motivation; this was notably evident during the pandemic when students had fewer opportunities to interact with their peers, which may reduce some students' motivation. Therefore, teachers should encourage students to interact more through different approaches, such as creating a friendly environment through information technology tools to facilitate student interaction (Ferrer et al., 2020). Ferrer et al. (2020) and Anderson (2008) found that online learning can play a mediating role and have a positive impact on both intrinsic and extrinsic motivations, allowing for more extensive educational communication among students. Thus, educational environments require a strong information technology infrastructure to foster student learning. The infrastructure should also have sufficient bandwidth and storage space to enable personalized learning.

Generally, the level of motivation is related to learning achievement. As such, teachers should consider how to enhance motivation through learning activities when designing course content. Some researchers suggest that integrating different strategic pedagogies and technological capabilities to promote students' intrinsic and extrinsic motivation, such as integrating augmented reality (AR) into the curriculum or using information technology tools to facilitate communication and interaction (Anderson, 2008; Ferrer et al., 2020; Khan et al., 2019; Smith et al., 2021). Therefore, the present study adopted different learning tools to integrate learning activities with the aim of attracting students' attention and promoting their motivation to learn. Moreover, the experimental design involved a different interactive e-learning platform that integrates various features to stimulate students' external motivation, such as interactive and pedagogical support modules.

2.2.2 Course Engagement

Student engagement has been found to be a critical factor that correlates with student learning success and satisfaction. Thus, how to effectively promote greater student engagement in individual courses is an important issue (Starr-Glass, 2020). Therefore, faculty should consider effective ways to increase engagement in their courses, which is strongly correlated with successful student outcomes. Research suggests that the learning environment may be a factor in triggering an impact on sustained interest and engagement, particularly in open-ended learning tasks where students need to understand course content while respecting their ideas and appreciating their contributions (Gomoll et al., 2016). Daniels, Goegan, and Parker (2021) found that students have goals and are engaged, which suggests that students can master knowledge and not do worse than their peers in order to develop student goals and engagement. Sadera et al. (2009) found that online learning is a way to promote interaction, collaboration, and community. These factors are vital to helping students succeed in their programs, which can positively impact academic achievement. However, the broad exploration of the nature of engagement is a complex issue that encompasses various facets, which can affect the level of student engagement. Sinha, Rogat, Adams-Wiggins, and Hmelo-Silver (2015) had similar findings, indicating that there are different facts in engagement. To further explore the reasons, the authors extended the existing engagement framework and formed a new framework with four facets: behavioral, social, cognitive, and conceptual-to-consequential forms; this is especially evident in the social and behavioral dimensions, both of which can increase students' consequential engagements (Sinha, Rogat, Adams-Wiggins & Hmelo-Silver, 2015). Overall, this study highlights that the concept of engagement is complex, and the authors suggest that the use of computer support can improve high-quality cognitive and conceptual-to-consequential engagement (Sinha, Rogat, Adams-Wiggins & Hmelo-Silver, 2015). Some researchers have suggested that student engagement in learning occurs not only in the classroom but also across the school; however, several factors influence engagement, such as cognitive, affective, and behavioral engagement (Cooper, 2014; Nguyen, Cannata & Miller, 2018). This indicates that engagement is a multi-factorial concept. It is not just a simple theory; it has various external or internal factors that also affect the level of student engagement in learning. More importantly, this theory has been shown to impact student learning outcomes and performance. Additionally, this paper suggests that the level of the concept of student engagement not only has a positive impact on school performance and persistence but also affects student performance in the classroom, such as providing challenging and authentic tasks (Fredricks, Blumenfeld & Paris, 2004). This highlights that the concept of student engagement is not only a single component. Therefore, it can integrate three factors, including behavioral, affective, and cognitive, because these factors are dynamic and not isolated processes for each individual.

As mentioned above, engagement is an important indicator that can be considered as a measure of learning performance. There is much research demonstrating that engagement is related to learning success and learning performance (Starr-Glass, 2020; Sinha, Rogat, Adams-Wiggins & Hmelo-Silver, 2015). However, other researchers have found that participation is a more complex concept that encompasses curricular engagement while further integrating more indicators to form a new framework; this can be derived by synthesizing two different papers, including behavioral, social, cognitive, affective, and conceptual-to-consequential (Sinha, Rogat, Adams-Wiggins & Hmelo-Silver, 2015). This also highlights the fact that there are differences in engagement levels and that engagement levels are more important to teachers because they should consider how to effectively integrate things of interest or learning strategies to capture students' attention and engagement. Therefore, the present study integrated e-learning platforms that promote student engagement in learning activities and include various features that encourage students to have more opportunities to use the e-learning platform for learning and interaction. We will analyze students' learning engagement through the instructional design proposed in this study.

2.3 Computational Thinking

In the digital age, technology has become the most critical skill worldwide. As such, courses related to Information and Communication Technologies (ICT), such as programming languages, data analysis, and algorithms, have become popular subjects. Therefore, computational thinking was proposed by Wing (2006), who argues that the broad definition of computational thinking is using computer science to solve problems and further explores it to include the design of systems and understanding of human behavior. Wing (2006) suggests that computational thinking not only provides information processing and basic mathematical skills but also helps students understand new content skills effectively (Fletcher & Lu, 2009). Some research suggests that the concept of computational thinking includes computer science problem-solving strategies, which indicate that learners can solve problems effectively, thus highlighting computational thinking as an essential skill (Google, 2015; Wing, 2006). Other scholars have suggested that everyone should have some degree of computational thinking because it involves programming skills and can be used in everyday life; thus, they encourage teachers to develop students' computational thinking skills more often through programming courses (Hambrusch et al., 2009; Israel et al., 2015). According to this discussion (Wing, 2006; Google, 2015; Israel et al.,2015; Hambrusch et al., 2009), the importance of computational thinking in education is more prominent, not only as a theory, but also can be used in daily life. Some researchers (Pinto-Llorente et al., 2016) have expressed that to actively promote computational thinking education, programming is a good way to cultivate it. According to the experimental results, programming was found to have a positive impact on improving computational thinking ability (Pinto-Llorente et al., 2016). Furthermore, to promote the cultivation of computational thinking talents, technology companies have designed a series of computational thinking learning materials. They have also indicated that learning computational thinking is a trend and one of the critical development skills for talents in related jobs (Google, 2015). Generally, The computational thinking of some researchers consists of several main elements: (1) decomposition, (2) abstraction, (3) algorithmic thinking, and (4) data analysis/representation (Csizmadia et al., 2015; Juškevičienė & DagienĖ, 2018; Csizmadia, Standl & Waite, 2019; Google, 2015). This is detailed below, and the process is presented in Figure 1.

1. Decomposition: A method of analyzing problems or large systems that can be separated into multiple components to make complex issues easier to solve and understand.
2. Abstraction: A way to make a problem or system easier, to reduce unnecessary details, and to have the correct information to use in developing the model.
3. Algorithmic thinking: The ability to consider consequences, including designing an algorithm for step-by-step problem-solving.
4. Data analysis/representation: This involves the process of data collection and description appropriately while organizing data into graphs, charts, or text.

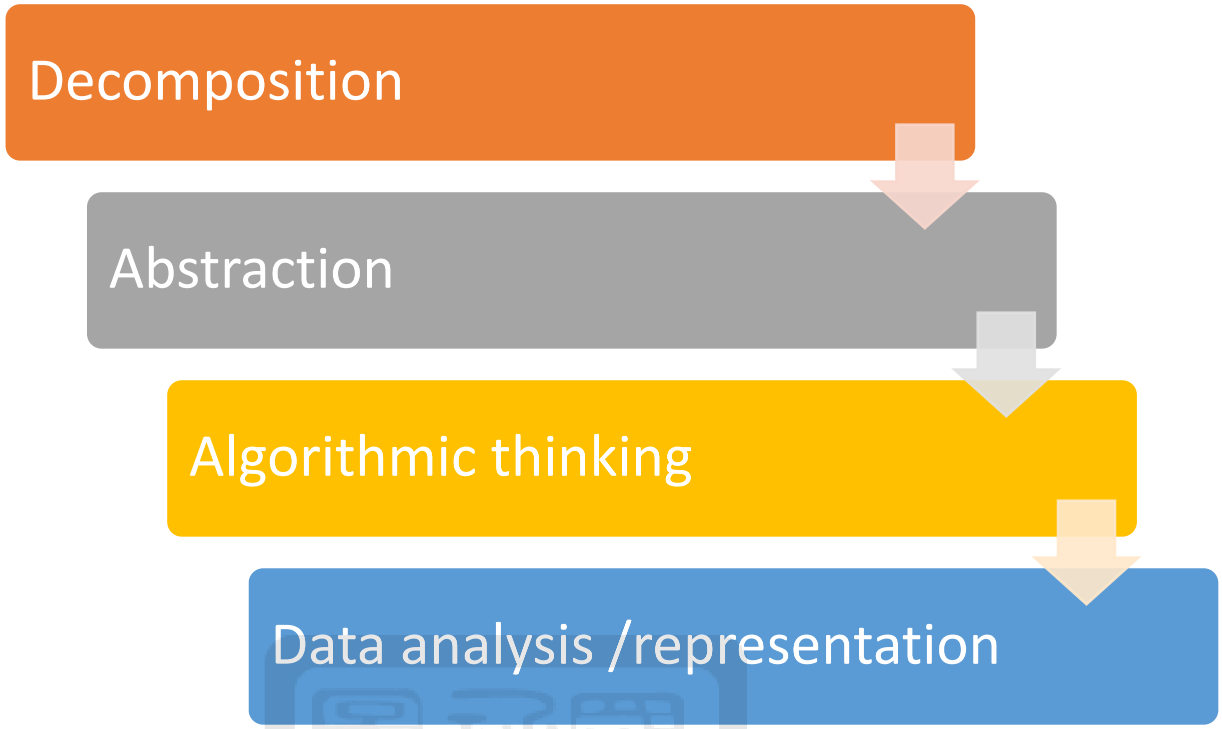

Figure 1. Computational thinking processes

Along with this discussion, the theory of computational thinking has been proposed and has attracted research attention (Wing, 2006; Google, 2015; Pinto-Llorente et al., 2016). Computational thinking is considered the ability to think about problem-solving and includes algorithmic thinking that makes problem-solving more effective. Additionally, computational thinking includes information processing skills and basic math skills that help students understand content skills more effectively. Thus, educators should be more proactive in promoting computational thinking skills in students. Research also suggests that developing computational thinking skills can be effectively developed through a programming language (Pinto-Llorente et al., 2016). Therefore, the present study proposes a series of experimental designs, each with a different interactive e-learning platform and different teaching strategies to develop students' programming skills to cultivate students' computational thinking skills through computer science.

2.3.1 Computer Science and Programming

The previous section highlighted that programming is an excellent way to develop students' computational thinking skills and has been shown to positively impact computational thinking (Pinto-Llorente et al., 2016). Therefore, computer science plays a crucial role in developing computational thinking skills, which also points to the fact that computer science not only develops students' ability to organize problems but also teaches them to understand more clearly how computers think and how they work (Dougherty, 2012; Giannakos et al., 2017; Harlow et al., 2018). Furthermore, a high level of competency in computer science can stimulate more innovative thinking while developing related skills such as project management, team building, software evaluation, progress monitoring, communication, etc. (Dougherty, 2012; Giannakos et al., 2017; Harlow et al., 2018). Although computer science issues have attracted the attention of many researchers, integrating computer science effectively is still a big problem. Falkner, Vivian, and Williams (2018) and Bell, Andreae, and Robins (2014) indicate that it is more challenging to integrate computer science into education, especially for elementary school teachers who do not have enough background knowledge or even experience in teaching related courses. This also makes it more challenging to teach programming courses. However, programming is a specialized and hierarchical skill. Thus, beginners will encounter many problems when learning to program, such as syntax errors or structural errors. As such, it is challenging for students to recognize the meaning of the compiler and solve it independently if they do not have sufficient knowledge (Rahmat et al., 2012). Furthermore, programming requires intensive practical programming, but teachers usually teach according to the course progression involving a time factor. There is no time to wait for students to fully understand, so students can not fully understand the course content, which leads to students not mastering the programming and problem-solving skills. Hence, learning results become unsatisfactory (Aleksić & Ivanović, 2016). To solve this problem, some researchers have suggested that block-based programming is a good approach for beginners; many learning tools design block graphical interfaces so that students can drag blocks and link to other blocks. Graphical interfaces help students to learn programming more easily, which also highlights that computer science learning has become a global phenomenon (Bau et al., 2017; Cooper et al., 2000; Fraser, 2015).

As mentioned above, computer science is considered a teaching strategy to develop students' computational thinking. Research has proven its positive impact on the development of computational thinking skills (Pinto-Llorente et al., 2016). Furthermore, computer science not only develops the ability to organize problems but also helps students develop related information skills such as project management, team building, software evaluation, progress monitoring, and communication (Dougherty, 2012; Giannakos et al., 2017; Harlow et al., 2018). The present study suggests that computer science should cover more programming areas, including opportunities for students to explore topics related to computer science, such as algorithms, artificial intelligence, formal languages, human-computer interaction, and cryptography. This indicates that computer science is a complex, comprehensive concept involving integrated competency that covers different professional disciplines (Bell, Andreae & Robins, 2014). To develop students' computer science skills, this study proposes various interactive e-learning platforms and integrates different learning strategies to help students learn programming languages. For example, the e-learning platforms integrate peer learning strategies and interactive features to promote students' motivation and encourage them to participate more in learning activities.

3 Research Methodology

Chapter 1 illustrated that the development of ICT literacy is a trend because it develops students' ICT-related skills, such as computational thinking, problem-solving skills, and computational science, as well as the use of ICT literacy in everyday life. However, how to effectively develop students' ICT literacy through programming is an issue that needs to be addressed. Some researchers have suggested that the reason students lack motivation to program is that they are not motivated to program because programming is relatively abstract and not easy for beginners (Aleksić & Ivanović, 2016; Lai, Jeng & Huang, 2020; Kelleher, Pausch, 2005; Linder, Abbott & Fromberger, 2006; Fletcher & Lu, 2009). Therefore, to solve this problem, the present study proposes two e-learning interaction platforms that integrate different teaching strategies into instructional designs. As such, there are two instructional study designs. This study aimed to observe the instructional design and explore which approach can help students achieve better learning outcomes. Each experimental design is described below.

3.1 Experimental Design 1: A Real-time Programming Teaching Support System

3.1.1 Participants

In this experimental design, we recruited 24 students to participate. They were all information management university students, meaning most had some programming knowledge and programming development experience. Therefore, the student participants had already studied programming-related courses such as web development, Java, and object-oriented programming. This implies that most of the students have had some learning experience in learning programming. Therefore, this study recruited these students as experimental subjects, with 24 student participants in this experiment. Consequently, we expected that our proposed instructional design would help students to achieve better learning outcomes.

The experiment lasted one semester. We collected students' midterm and final exam results and questionnaire data. We analyzed students' learning performance and motivation questionnaire data after the experiment and explored the impact of the learning experiment.

3.1.2 The Proposed Learning System Modules

This study proposed a real-time programming support system to support students in learning programming when they perform learning activities. The system integrated a real-time response module, an exception detection module, a user collection module, and a filter module. As such, the system could continuously monitor students' error messages and provide suggestions to students as a reference when they encounter error messages. The four modules are described as follows.

Exception detection module: This module automatically monitors students' programming logs as they write code until an error message occurs in the compiler.

User collection module: This module integrates an exception detection module. The exception module detects error messages, and the module can collect the student's error messages and store them in the database. Figure 2 is presented below.

Real-time response module: This module has two functions: the real time and the response function. The module is connected to the exception detection module, which provides students with real-time suggestions for learning programming references. Students must understand the means and try to solve the problems so the program students can learn programming in more depth, which is shown in Figure 3 below.

Filter module: This module can filter the error messages because the programming is stacked. Since error messages will print all stacked error messages, it is difficult for students to read the error messages. To solve this problem, the module will aggregate the error messages and filter the unnecessary errors to help students find the main error and resolve it more easily.

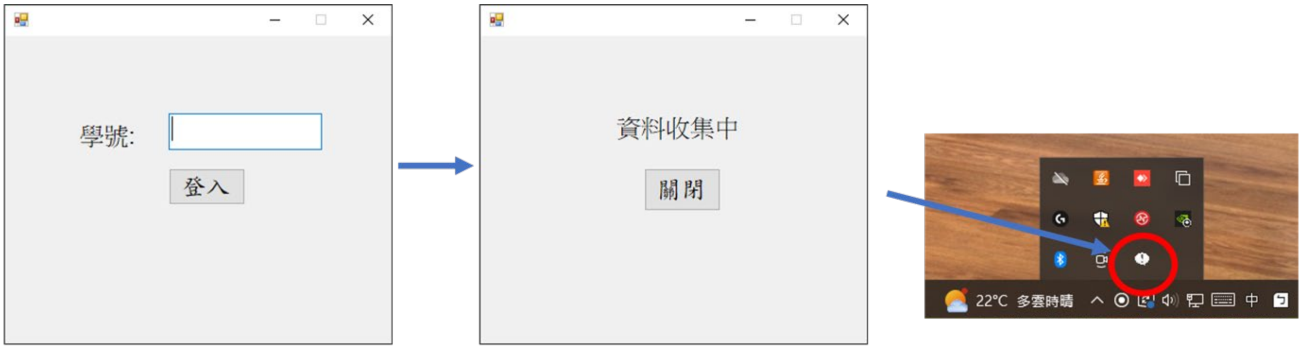

Figure 2. User collection module

Figure 3. Real-time response module

3.1.3 Experimental Design and Teaching Strategies

This study proposed a learning system. According to the proposed learning system module, the system can provide students with real-time support information as teaching assistance information. Therefore, students can obtain more information related to error messages, which can reduce the learning burden of students. In programming development, it also facilitates students to solve their own problems. Additionally, we developed students' computational thinking skills through computer science. The curriculum design was created by referring to the article "Computational thinking-A guide for teachers," in which some computational thinking strategies were highlighted as teaching methods. The curriculum design of this study is presented in Figure 4. The curriculum design generally included three steps: decomposition, abstraction, and algorithmic thinking. However, we do not include data analysis/representation. The learning activities did not arrange data output and analysis steps because it is challenging to measure the students' data analysis/representation as follows. The curriculum design details are described below.

Let us use this problem as an example to write a program to detect text entered by the user's keyboard. Where the user presses 'A' to indicate that the screen displays 'hello' and 'B' to indicate 'bye-bye.'

(1) Decomposition: Students will learn how to decompose the problem. This problem has four steps: (1) detect the keyboard, (2) detect the button press, (3) the screen displays the result, and (4) the programming will continuously detect the user's keypress.

(2) Abstraction: Students will learn abstraction skills and how to use them in programming

learning based on the results of the decomposition steps and using them in the abstraction steps. This problem has four steps: (1) create a scanner object to detect the user's input, (2) use conditions to determine the text entered, (3) set the corresponding text, and (4) use loops to achieve the effect of continuous programming.

(3) Algorithmic thinking: Use the results of the abstract steps and implement them. Integrate the conditions, loops, corresponding text, etc., into the function.

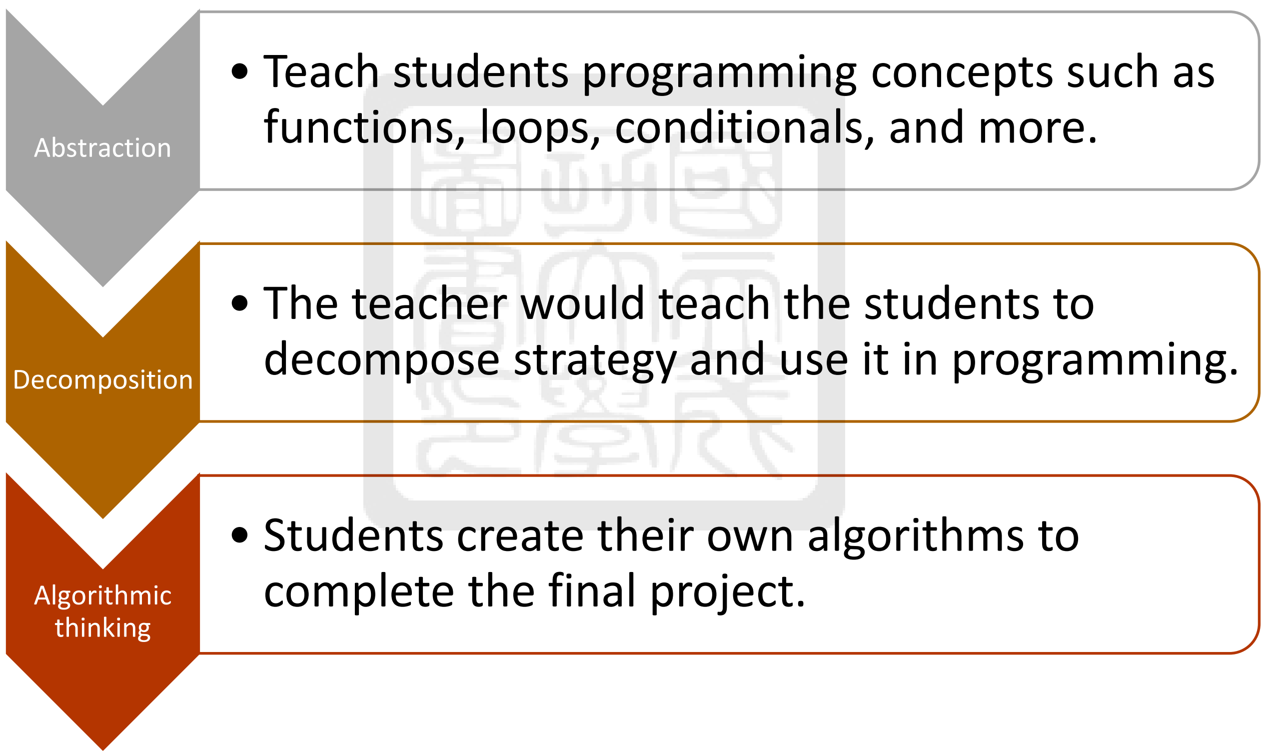

Figure 4. Computational thinking in the course

Regarding instructional strategies, this study incorporated scaffolding theory into the instructional design, which was proposed by Vygotsky (1978). The author suggest that the actual level of development and the level of potential development are zones of proximal development (ZPD)

Vygotsky (1978). Scaffolding theory can develop students' ability to solve problems and can actively promote their own possible development. On the other hand, researchers have also suggested that teachers provide supportive scaffolding to train learners to complete learning tasks independently until they have a certain level of competence, gradually reducing this support (Margulieux & Catrambone 2021). Based on this view, the present study integrated scaffolding theory teaching strategies. Therefore, the learning activities were designed in three stages. A detailed description and flow chart are presented in Figure 5.

Step 1: Before the midterm exam, the teacher arranged more simple programming introduction chapters to provide students with time to understand the programming and development environment. This may prevent some students from being exposed to different programming languages for the first time, affecting data accuracy.

Step 2: The teacher set up various programming-related problems for students to practice. Students completed the program development with the help of the system during the learning activities, which aims to allow students to learn higher-level programming background knowledge. Additionally, the system can collect practical programming data and automatically store error information in the database.

Step 3: At the end of the experiment, the teacher arranged a final project to analyze the changes in student learning performance and explore the students' frequent error messages.

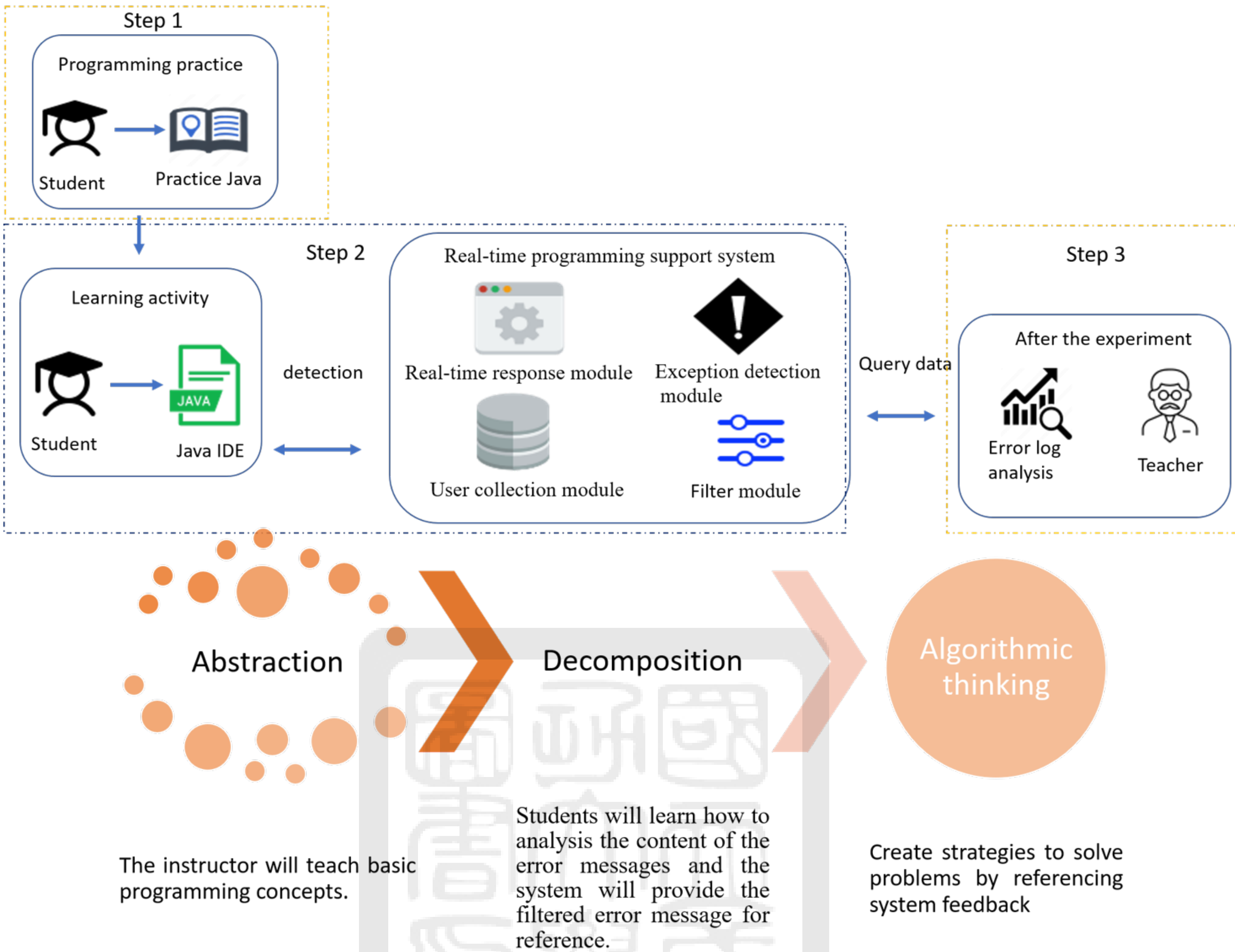

Figure 5. Experimental design

3.1.4 Research Instruments

To explore students' learning performance, midterm and final exams were collected and used as pre-tests in this study. Since students were just introduced to the new programming language and did not start using the system proposed in this study, we used step 1 as the pre-test data. Then, step 2 defined the experimental phase in which the students had various learning activities. The real-time programming support system was run as a backend application to collect students' error messages. Then, the teacher analyzed and filtered the error messages after the experiment. They explored which error messages appeared most frequently when the students did the programming implementation exercises. Moreover, we set the final exam as the post-test. Thus, at the end of the experiment, we had both pre-test and post-test data and then performed paired t-tests to analyze changes in student performance.

3.1.5 Learning Performance Measure

To further explore students' academic performance in computational thinking, the students' midterm and final exam scores were collected to examine changes in learning. In this study, grades were analyzed as academic performance because the questions in the midterm and final exams were designed to include computational thinking concepts such as abstraction, decomposition, and algorithmic thinking. The exam questions were divided into two parts. The first part measured the students' basic concepts. The second part was the programming implementation, where the students submitted their exam answers and uploaded the complete code and snapshots so that the teachers could measure the students' computational thinking through the midterm and final exams.

3.1.6 Learning Engagement Measure

In this experiment, there were many learning activities for students to practice and learn. As such, we collected students' learning engagement activity logs to analyze the level of student learning engagement. Students who participated in the learning activities scored 1; those who did not participate in the learning activities scored 0. Therefore, teachers could use the learning engagement activity logs to count students' learning engagement.

3.2 Experimental Design 2: Note-taking Interactive E-learning Platform

In this experiment, we further investigated the effects of students' different interactive e-learning platforms on programming and participation in learning performance. Thus, a note-taking interactive e-learning platform was proposed for data collection and applied to the teaching curriculum. Additionally, to enhance students' motivation in programming learning, this experiment incorporated peer learning theory into the learning activities. The details are described in the next section.

3.2.1 Participants

A total of 59 junior students were recruited for this experiment. Most were from the Department of Information Management, aged 21-22, from one of the universities in Tainan. This course regarded the Internet of Things, which means it differs from the traditional programming course. It requires both a programming language and software background knowledge, which students must have in order to begin the learning activities. Therefore, we recruited the students as experimental subjects because they had relevant experience in programming-related development.

The experiment lasted one semester. We collected students' midterm and final exam results and questionnaire data. We analyzed the student learning performance and motivation questionnaire data after the experiment and explored the impact of the learning experiment.

3.2.2 Note-taking Interactive E-learning Platform System Module

This experiment proposed a note-taking interactive e-learning platform system that aims to enhance students' motivation to learn through the system. Moreover, the system integrated learning activities as the instructional design based on the peer learning theory. We combined the instructional design and peer learning theory to form the experimental design of this experiment. Some research indicates that peer learning increases opportunities to share ideas, which not only develops students' thinking skills but also has a positive impact on learning achievement

(O'Donnell & King, 2014; Zou et al., 2012). Furthermore, several researchers have demonstrated that peer learning is an effective teaching strategy that is more beneficial than traditional learning in that students have more opportunities to practice and learn in a peer learning environment (Boud et al., 2014; Fisher & Frey, 2019; Meschitti, 2018). As such, this experiment's instructional design incorporated peer learning strategies into learning activities to attract students to use it more willingly. To increase the opportunities for peer interaction, the system was designed with features such as source sharing, notification, note-taking, etc. The details are described below.

Note-taking module: The note-taking module provides a note-taking learning environment for students to write down their notes. Research has indicated that note-taking is an information process that requires not only a high level of attention but also that students must internalize the information, understand it, and absorb the knowledge before writing it down (Bui et al., 2013; Chang & Ku, 2015). Based on these insights, the module provides note-taking features, including writing notes, inserting graphics, drawing tables, etc. Students can more easily write any notes they want to make in their own way. The note-taking module is presented in Figure 6.

Social module: This module was designed with some interactive features. To attract students to use it, the design refers to some social media; it integrates the Like button, resource sharing, notifications, etc. Its purpose is to encourage more sharing and interaction through the social module. The social module is shown in Figure 7.

Upload module: This module provides teachers with a feature that supports uploading materials in multiple formats. This means that teachers can upload different material content, such as graphics, tables, links, videos, etc., according to the needs of the course. The upload module is presented in Figure 8.

Popular article module: This module categorizes the content of the user's notes and displays the calculation results in the system so that students can quickly find the notes they need. This popular module is presented in Figure 9.

Source sharing module: This module provides a sharing feature for other students. This resource is from students taking notes. The notification module notifies the student that someone wants to share it with them. Additionally, this feature is essential for learning because the module includes social tagging, which means that learners leave some information for other peers to refer to; receivers of the information can understand the content more easily and thoroughly (Klašnja-Milićević et al., 2018).

Notification module: This module's function is to notify users that someone will share a resource with them so students can quickly view the shared content through the module. The notification module is presented in Figure 10.

Figure 6. Note-taking module

Figure 7. Social module

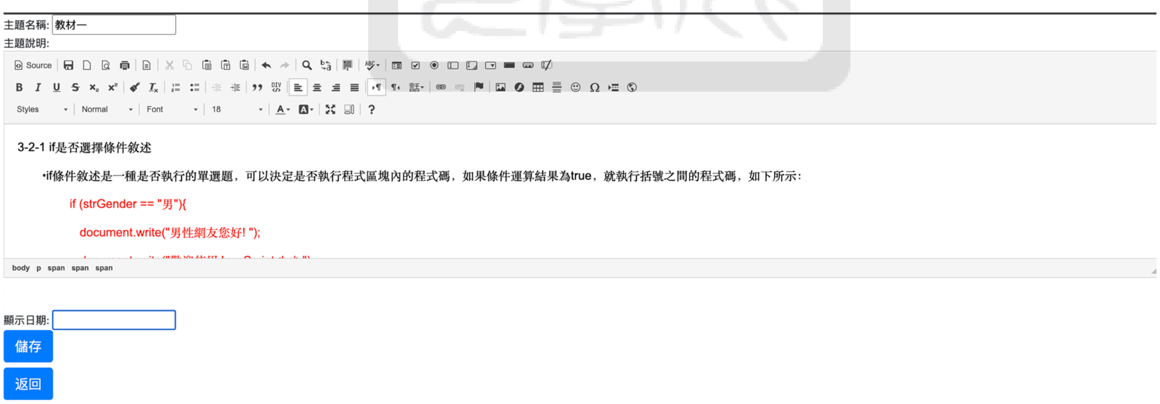

Figure 8. Upload module

相關筆記

Show 10 entries Search:

建立者	筆記內容	建立日期	點閱率	前往筆記
林尚豫	讓使用者...	2019/3/26 下午 04:51:15	0	前往筆記
林尚豫		2019/4/23 下午 05:05:35	0	前往筆記
林尚豫		2019/4/30 下午 05:08:57	0	前往筆記
林尚豫	叫出彈跳...	2019/3/19 下午 04:41:33	0	前往筆記
林尚豫	用css...	2019/3/19 下午 04:42:36	0	前往筆記
林尚豫	一次設定...	2019/3/19 下午 04:46:27	0	前往筆記
林尚豫	尋找"r...	2019/3/19 下午 04:47:52	0	前往筆記
林尚豫	尋找"b...	2019/3/19 下午 04:48:40	0	前往筆記
林尚豫	這裡的t...	2019/3/26 下午 04:52:10	0	前往筆記
林尚豫		2019/3/26 下午 04:55:40	0	前往筆記

Showing 11 to 20 of 77 entries

Previous 1 2 3 4 5 ... 8 Next

新增筆記 回首頁

Figure 9. Popular article module

Figure 10. Notification module

3.2.3 Learning Activity and Teaching Design

This experiment arranged a series of learning activities in the curriculum to enhance students' motivation and learning performance. The learning activities incorporated different teaching strategies, such as peer learning and note-taking theory. Peer learning strategies have been proven beneficial in improving student learning outcomes, accelerating learning, and facilitating interaction between students and their peers (Boud et al., 2014; Fisher & Frey, 2019; Meschitti, 2018). Some researchers have suggested that team members are the most crucial in peer learning because they work together with their peers to complete learning tasks, support each other, and use each person's abilities to work together to complete learning tasks (Chen et al., 2012; Shih et al., 2018). On the other hand, to achieve better learning outcomes, this experiment incorporated note-taking instructional strategies into the curriculum instruction, which differs from traditional note-taking strategies because students must write their own notes that teachers do not provide. This means that students should understand the content of the curriculum and then write it down in their own words. According to Heitmann et al. (2018), thinking and cognitive effects are critical competencies for note-taking skills; therefore, teachers should encourage students to write their own notes, promoting cognitive effort and higher thinking skills. In the course design, we adopted the same approach as the Experiment 1 course design. Thus, teachers provided exercises to teach students how to use computational thinking strategically to solve problems. See Figure 4 for details of the computational thinking instruction.

The previous discussion explained the concepts of note-taking and peer learning. Research has proven that these teaching strategies are effective and beneficial to learning. Therefore, we combined the theory of note-taking and peer learning in the experimental lesson. During the experiment, students used the note-taking interactive e-learning platform to conduct learning activities with a series of features designed specifically for the experiment. First, the teacher would teach the class according to the schedule. Each student had their own account in the notes interactive e-learning platform system. The teacher asked the students to write notes on the system.

There was no limit to what they could write, such as programming practice records or problem-solving processes. Subsequently, students were randomly divided into groups. Members of the same group could read the group's notes. Students could use the social module to interact with their peers and share helpful resources to aid peers in solving problems quickly. Finally, the teacher collected and analyzed the data to explore changes in students' performance and motivation to learn. The flow diagram of the learning activities is presented in Figure 11.

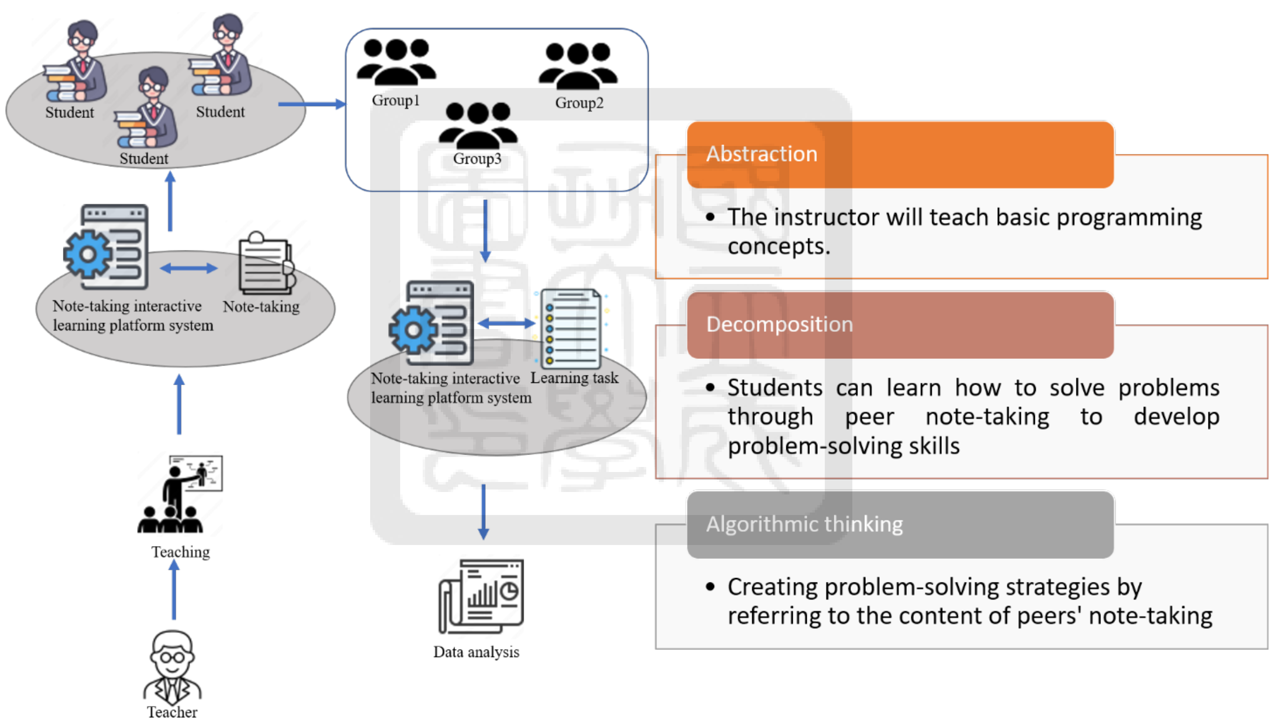

Figure 11. Learning activity design

3.2.4 Data Acquisition and Analysis

Two types of data were collected in this experiment, including exam scores and questionnaires. Regarding test scores, we collected students' midterm and final exam scores as a pre-test. Since students had not yet started using the proposed system and teaching methods, this time span was defined as a pre-test. Then, we integrated the teaching strategies and the note-taking interactive e-learning platform system into the curriculum so that the final exam scores could be used as the post-test. Regarding the questionnaire, the Instructional Materials Motivation Survey (IMMS) was used to measure students' motivation and explore the changes in students' motivation. The IMMS has a high-reliability coefficient of 0.96 (Keller, 2006). This indicates that the questionnaire has high reliability, so we used the IMMS questionnaire as a measurement tool. The IMMS questionnaire was developed from the ACRS motivational model, which was developed by (Keller, 1983; Huang et al., 2006). This model has four dimensions, including attention, relevance, confidence, and satisfaction, as described below.

Attention: Instructional strategies should attract students' attention, which affects information processing and guided comprehension.

Relevance: Learners recognize a topic's relevance to a need or goal, making learners more willing to learn.

Confidence: In understanding a topic that builds student confidence, learners have confidence that they can succeed; they are more willing to engage in learning activities.

Satisfaction: Learners should receive some type of reward, such as a sense of accomplishment, entertainment, etc., that will help learners feel satisfied with the learning experience and continue to learn.

The IMMS scale was used in this experiment, which has 36 questions and uses a 5-point Likert scale for measurement: 1 point = "strongly agree" and 5 points = "strongly disagree." The instructor arranged some time for the students to complete the questionnaire before the end of the experiment.

3.2.5 Learning Performance Measure

This study used midterm and final exam results as academic performance measures. We used the midterm results as a pre-test because students learned basic programming concepts but did not use the teaching strategies proposed in this study. After the midterm exam, students started to use the teaching strategies proposed in this study, including the note-taking interactive e-learning platform and peer learning teaching strategies. We used the final exam as the post-test to measure students' learning performance. To further explore the students' learning performance, the teacher designed a series of questions for the students during the examination, such as basic concepts and programming implementation. Therefore, students were required to upload the program code and present a snapshot of the execution result to the compiler. Based on the exam results, the teacher could measure student learning performance and the performance of students' computational thinking skills.

3.2.6 Learning Engagement Measure

In this study, a series of learning activities were organized for students' learning and practice to assess the level of students' participation in the learning activities. We collected students' learning activity logs when they participated in the learning activities. Students were scored 1 if they participated in the learning activities; those who did not participate in the learning activities were scored 0.

4 Results and Discussion

4.1 Experimental Design 1: A Real-time Programming Teaching Support System

4.1.1 Discussion 1: Does the Real-time programming teaching support system improve student learning performance?

This experiment used scaffolding theory as the teaching strategy. The teacher arranged various learning activities for students to practice and become familiar with programming. Then, the teacher arranged advanced learning activities to develop the students' programming background knowledge. This means that some students were less involved in learning activities, and they had less access to the real-time programming support system, which affects the accuracy of data analysis. Therefore, a smaller number of students with no less than 70% participation were filtered to ensure data validity, which was eventually collected from 22 students. Table 1 illustrates that the results indicate a significant improvement in learning performance, with a score of 67.32 in the pre-test phase; the integration of the experimental design into the curriculum improved the score to 73.14. This suggests that the experimental design and real-time programming instructional support system proposed in experiment 1 of this study had a positive impact on programming learning. The standard deviation of students in the class decreased. This is because the students received real-time help information and filtered out unnecessary error information, which reduced the learning burden of the students and enhanced their independence in completing the learning tasks.

Su and Chen (2018) and Chien et al. (2015) had similar results that also proposed a real-time system and applied it to curriculum activities. This not only positively affected student engagement but also influenced student discussion processes and conceptual learning outcomes.

As such, the authors suggest that teachers should consider designing real-time response displays based on student learning, as too much display content can distract or mislead students. The critical function of the real-time response system as a learning tool effectively improves students' conceptual understanding and knowledge construction, makes students feel novel when using new technologies, and makes them work harder to learn (Shyr, Hsieh & Chen, 2021). This study integrated live chat into the teaching environment as an online help tool, which provides real-time, easy access to real time helps for students. Thus, the live chat feature makes online students feel satisfied with their learning and feel more cared for by the teaching team (Broadbent & Lodge, 2021). Overall, the experiment was effective in improving students' learning performance. This is primarily because the system integrated real-time features, which provided students with real-time learning assistance and filtered error messages. It also reduced the stress of learning programming.

Table 1. Class Score Results

	Pre-test		**Post-test**			
	Mean	**S.D.**	**Mean**	**S.D.**	**t-value**	**p-value**
Test score(N=22)	67.32	14.917	73.14	13.868	-2.189	0.04*

*p < 0.05

4.1.2 Discussion 2: Does the Real-time programming teaching support system improve student learning engagement?

In this experiment, we designed many learning activities for students to practice and study as assignments. Then, students were required to submit the assignments on the platform. We counted the students' assignments to measure their learning engagement. However, before the midterm exam, we did not use the "real-time programming teaching support system" and teaching strategies proposed in the experiment, so we defined this phase as a pre-test. Figure 12 illustrates four assignments (assignments 1 to 4) as the pre-test data. The collected data indicates that the students' engagement ranged from 0.77 to 0.81, which suggests that most students had a good engagement in learning before the experiment started. Then, the experiments were integrated into the course, and data were collected as post-tests. After the final exam, the instructor sorted and analyzed the data. Figure 12 illustrates four assignments (assignments 5 to 8) as the post-test data. The results indicate that most students were more engaged than in the pre-test, with engagement scores ranging from 0.72 to 0.95 for learning activities. This finding suggests that the experiment helped students to increase their engagement in learning so that most of them were more willing to participate in learning activities than before the experiment.

However, to further explore the changes in student engagement, we used a paired t-test to measure the pre-test and post-test changes. The results are shown in Table 2 below. Table 2 indicates that class students gained 0.77 engagement in the pre-test and class students gained 0.88 engagement, a total increase of 0.11. However, no significant difference was achieved. This experiment had a specific effect in helping students to become more willing to engage in learning activities. As such, the class improved by 0.11 even though no significant difference was reached at the end of the experiment. The presumed reason is that the real-time response module provided appropriate assistance to students when they had error messages. However, the response messages were not clear and specific, which made students confused. The scholars also mentioned that the proposed content of the response should include two elements: (1) be precise and clear; (2) avoid inconclusive or misleading content (Price et al., 2010). However, the experiment with real-time features also helped increase students' learning engagement. According to Zeinstra et al. (2023), real time is an essential feature of the learning process because teacher-student interactions can link macro-levels, features, and outcomes; it is also a factor that affects student engagement and learning atmosphere. Therefore, teachers should pay more attention to real time interactions between teachers and students. Furthermore, this study proposed an adaptive immediate feedback system that integrates a programming environment to help students learn programming. The system provides personalized positive and corrective feedback to students. Subsequently, the experimental results indicate that students not only achieved better learning performance but also increased student engagement in learning. This is because the system provides immediate feedback to make students more focused on learning and more attentive to persist in their computer science courses (Marwan et al., 2020). As a result, we recommend that real-time functionality be integrated into the learning environment, which has proven to be a critical factor in influencing student performance and enhancing engagement in learning activities.

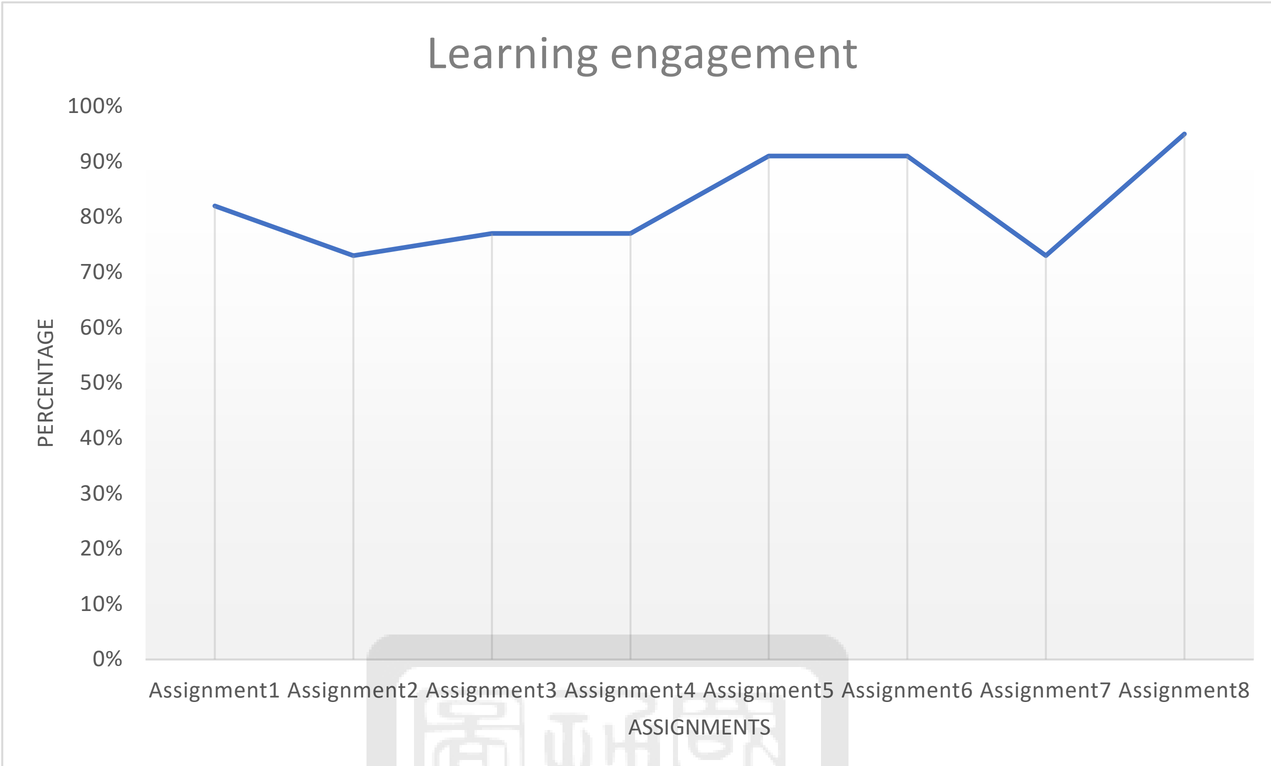

Figure 12. Learning engagement

Table 2. Learning engagement (n=22)

	Pre-test		**Post-test**			
	Mean	**S.D.**	**Mean**	**S.D.**	**t-value**	**p-value**
Test score	0.77	0.421	0.88	0.333	-1.824	0.07

According to Experiment 1, we found that most students had questions about the compiler error message. However, some of them preferred to discuss it with their peers, which created some problems because their peers may not have had enough background knowledge to understand the misconceptions, or the misconceptions implied that the students did not learn the correct concepts.

This study also illustrates that the process of student discussion can help in learning but only on the topic of solutions; peer learning approaches can have different learning impacts depending on the subject being studied. Therefore, we recommend that more detailed explanations of some common misconceptions be provided, which can also help students to learn more effectively.

4.2. Experimental Design 2: Note-taking Interactive E-learning Platform

4.2.1 Discussion 1: Does the note-taking interactive e-learning platform improve student learning performance?

This experiment proposed a note-taking interactive e-learning platform and peer theory and integrated it into the course. Therefore, students could use the platform during the course. The teacher arranged a series of learning activities, including code exercises, sensor applications, peer discussions, etc. After the experiment, the teacher analyzed the changes in student learning performance. We used the same criteria as Experiment 1 (70% engagement above) to filter the invalid data. This is because students with less than 70% engagement indicate that they typically used the note-taking interactive e-learning platform and teaching strategies proposed in the experiment less frequently. Not filtering the data would reduce the data reliability. We filtered seven students, so there were 52 students engaged in the experiment. Table 3 shows that the students scored 70.73 on the pre-test, which means that the students' class performed well. Subsequently, we incorporated the interactive e-learning platform and teaching strategies into the course, and the score was 79.52, a significant increase in learning performance. This indicates that the interactive e-learning platform and teaching strategies proposed in this experiment effectively improved learning performance. This is because the platform provides many interactive features for students to use and learn. Therefore, it is easier for students to share information and notes, which means that group members can read different notes and information from their peers. The system also provides various formats for students to take notes so that students can create their own unique notes and share them with peers. Team members supporting each other is a critical factor affecting the success of cooperative learning because team members can use each other's abilities to complete learning activities (Chen et al., 2012; Shih et al., 2018). Furthermore, note-taking is also a factor that affects student learning. Researchers also suggest that note-taking plays a critical role in helping students gain knowledge, learn, and succeed in their courses while improving students' understanding of content (Salame & Thompson, 2020). On the other hand, note-taking is a complex process. Before writing, students must be highly attentive to the information they receive and internalize and understand it through personal cognition (Bui et al., 2013). The note-taking platform is an excellent way to improve students' learning outcomes. Students must understand the content first and write down the notes in their own language. Students must internalize the content of the material, which develops students' learning ability and comprehension skills. Moreover, the system has a peer interaction feature that allows students to share their notes; group members can read their peers' notes, and students can refer to other notes to solve problems.

Table 3. Class Score Results

	Pre-test		**Post-test**			
	Mean	**S.D.**	**Mean**	**S.D.**	**t-value**	**p-value**
Test score(N=52)	70.73	14.055	79.52	4.804	-4.971	0.00**

*$p < 0.05$

**p <0.01

4.2.2 Discussion 2: Does the Note-taking interactive e-learning platform improve student learning engagement?

This experiment integrated the note-taking interactive e-learning platform and peer theory into the curriculum so the teacher could arrange learning activities to facilitate student discussion and interaction. To explore the variation in student engagement, we used the same criteria as Experimental Design 1: A Real-time Programming Teaching Support System. As such, we defined assignments 1 to 4 as the pre-test because the experiment had not started; students had not used the note-taking interactive e-learning platform and instructional strategies during this period. The experiment was integrated into the curriculum after the midterm exam, defined as the post-test phase. Figure 13 shows the results that most students had high engagement in learning activities, with scores ranging from 0.92-0.94. After the experiment, the scores ranged from 0.90-1, which suggests that most students had high engagement in learning. However, one of the assignments showed a decrease in engagement. To explore the changes in engagement, we used paired sample t-test to measure the pre-test and post-test changes, and the results are presented in Table 4 below. According to Table 4, the pre-test class student engagement score was 0.94; the post-test class student engagement score was 0.96, a slight increase of 0.02 and no significant difference. Two possible reasons exist for the slight increase in engagement in this experiment. First, at the pre-test stage, the engagement score was 0.94, which is very close to 1 if the engagement was still increasing, and it is challenging to observe significant changes. Second, the note-taking interactive e-learning platform provides a convenient feature for students to interact with their peers, such as sharing and reading each other's notes. Therefore, students had access to different notes to refer to when they practiced learning activities. This study has stated that technology devices integrated into the curriculum are an effective teaching strategy because they allow students to quickly answer questions posed in class and facilitate interaction among students. The experimental results indicate that technology devices not only have a positive influence on collaborative learning but also increase engagement in learning activities (Blasco-Arcas et al., 2013). On the other hand, a significant benefit of the note-taking function is that students become more actively involved in the lecture, as they must listen to and write down essential knowledge from the lecturer. This also has a positive impact on understanding when students are more actively engaged in learning activities (Bohay et al., 2011).

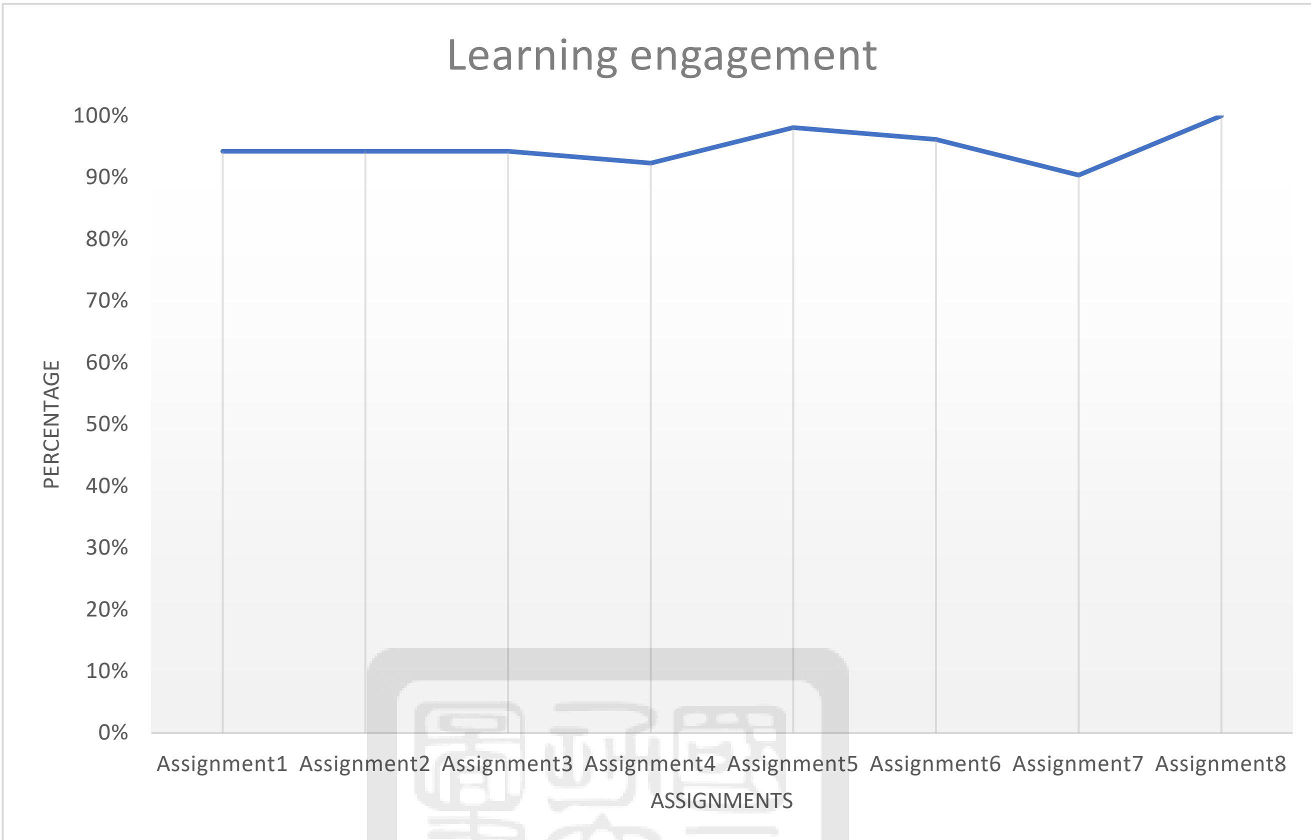

Figure 13. Learning engagement

Table 4. Learning engagement (n=52)

	Pre-test		**Post-test**			
	Mean	**S.D.**	**Mean**	**S.D.**	**t-value**	**p-value**
Test score	0.94	0.243	0.96	0.193	-1.293	0.197

4.2.3 Discussion 2: Does the note-taking interactive e-learning platform improve student learning motivation?

Learning engagement has an indirect effect on internal and external motivation, which is related to learning performance (Wu et al., 2020). The study (Guo, 2018) indicated that adolescents' motivation for academic autonomy was positively correlated with basic psychological needs satisfaction, life satisfaction, and engagement in learning. Thus, learning engagement could also be used to predict their effects on life satisfaction and engagement in learning (Guo, 2018). However, Table 4 illustrates that it was challenging to observe significant differences. As such, we also used the IMMS scale to measure students' motivation to learn. We used a one-sample t-test to calculate the results of the IMMS scale and referred to Atinafu (2021) by selecting 3 as the threshold value. When the score was higher than 3, it indicates that this experiment proposes that note-taking interactive e-learning platform integrated teaching strategy had a positive effect on learning motivation. The results are presented in Table 5. The attention scores ranged from 3.65-4.17, which indicates that students paid more attention to the experiment since it designed different features to attract students to use and learn. Therefore, students spent more time on the learning activity. The relevance scores ranged from 3.87-4.38, which indicates that students believed that the course was highly relevant to their needs. In addition, we used peer learning as a teaching strategy. As such, students could interact not only in the classroom but also with their peers on the platform. Therefore, students could learn from their peer notes. The confidence scores ranged from 2.6-4.04; most students had high confidence since this experiment used a peer learning strategy in which students could collaborate on learning tasks and final projects by referring to other peers' note or discussions with peers. However, we found that one of the questions scored very low at 2.6. This indicates despite the high attention and relevance of the curriculum design, the curriculum included code exercises, problem solving, which made students feel stressed and more challenged. Therefore, teachers should adjust the course level appropriately to reduce psychological barrier, especially learning stress, which can affect students' willingness to learn and their engagement in learning. Psychological problems are important factors that affect learning and influence learners' behavior and interactions in the environment. When students lack self-confidence and anxiety, this can impact learning performance (Fitriati, 2016; Wulandari et al., 2022). The satisfaction scores ranged from 3.96-4.25, which indicates that students considered themselves very satisfied with the design of this instruction and had a sense of accomplishment, especially after completing the final project.

Table 5. Four Dimensions Analyzed Results (N=52)

	Mean	**S.D.**	**p-value**
Attention	3.94	0.832	0.00
Relevance	4.09	0.781	0.00
Confidence	3.50	0.863	0.02
Satisfaction	4.07	0.718	0.00

According to Experiment 2, note-taking interactive e-learning platform provided students the ability upload their own note-taking. When we further explored the note-taking content, we found that part of student note-taking lacked organized content. Thus, other peers could not effectively learn knowledge or determine solutions to problem based on the notes of their peers. Some researchers have suggested that it is important to understand the impact of the note-taking approach on student learning. However, is challenging to effectively measure the quality of students' note-taking, especially as sometimes students do not take notes on things such as videos or diagrams (Witherby & Tauber, 2019).

Chapter 5. Conclusions

Information and Communication Technologies (ICT) are essential knowledge for the digital age, and everyone uses them in their daily lives. ICT literacy skills have become popular in recent years. Therefore, to improve citizens' ICT literacy, many governments are actively offering ICT-related courses such as computer science, artificial intelligence (AI), and the Internet of Things (IoT). These not only use ICT but also develop students' ICT literacy and computational thinking skills. As such, these courses have high learning value. However, despite the high learning value of programming courses, these skills are not easy for novices to learn because they require some basic knowledge, without which novices cannot recognize compiler error messages. Furthermore, due to the limited time available for courses, teachers typically do not have enough time to wait for all students to fully understand the content, making it difficult to develop students' ICT literacy. On the other hand, students must have basic knowledge and use it during intensive exercises, which also highlights the difficulty of learning programming. These learning barriers can make students uninterested in learning programming and while decreasing their learning engagement in these courses.

To address these issues, the present study designed two experiments to explore the changes in student learning performance and learning engagement. The first experiment integrated a real-time programming teaching support system as a teaching tool with scaffolding theory to help students learn programming, forming a new teaching strategy. The second experiment integrated a note-taking learning platform and peer learning theory into the learning activities, forming a new instructional strategy. The study results can be used as a reference for instructional design and learning activity design by instructors related to instructional programming in the future. Based on these experiments, we have obtained two research results, which are described below.

5.1 Experimental Design 1: A Real-time Programming Teaching Support System

This experiment incorporated a real-time programming teaching support system and scaffolding teaching strategy into the curriculum as an instructional design. The system can collect error logs and filter information when a student makes an error message. This reduces the learning burden on students who can build knowledge as they complete their own learning tasks. Table 1 shows that the real-time programming teaching support system and scaffolding were effectively able to improve student learning performance, and there was a significant difference (67.32 to 73.14). This finding indicates that the real-time programming teaching support system with scaffolding is an effective instructional design. It has real-time responsive features that facilitate building students' programming and computational thinking skills, which implies that students' ICT literacy and computational thinking skills were improved. Additionally, we explored the variation in students' engagement with this instructional design. Table 2 illustrates that the instructional design has an impact on learning engagement and, therefore, students had the willingness to learn, though it did not reach a significant level of difference. The reason is that real-time response messages may be confusing or unclear to students. Therefore, while real-time response messages may be helpful for programming development, not all students fully understand the content and means of real-time response messages.

5.2 Experimental Design 2: Note-taking Interactive E-learning Platform

This experiment incorporated a note-taking interactive e-learning platform and peer learning as an instructional design into the course. Students were randomly divided into groups. The platform provided an interactive environment for students to create their own notes and share them with their peers through different types of media, pictures, etc. Since group members had access to notes from peers, students could solve problems by referring to these notes. The results in Table 3 show that there is a significant difference in student learning performance. In the pre-test compared to the students got better learning; the students' score increased from 70.73 to 79.52. This indicates that the students gained ICT literacy and computational thinking abilities. This finding suggests that this platform and peer learning is a good way to promote student performance, especially peer learning. The system also has integrated interactive features that allow students to not only discuss in class but also write notes and interact with other peers. Moreover, we explored the changes in student engagement on the instructional design. The results showed a slight increase in student learning engagement (0.94 to 0.96) and fairly high scores in the pre-test. However, it was challenging to observe a significant difference in student learning engagement.

Chapter 6. Limitations and Future Work

This study proposed two experiments. The limitations of these experiments are described below.

6.1 Experimental Design 1: A Real-time Programming Teaching Support System

This experiment included only 24 student participants. We expected to select about 50-70 students as experiment participants. As such, the experiment results could be more accurate when the number of participants increases. Therefore, this course will recruit more students for future work to collect more data and analysis. Regarding function, there are two future works: (1) integrated machine learning and (2) emotion detection, the details of which are described below.

Integrated machine learning: Real-time programming teaching support systems can incorporate machine learning techniques. Scholars have indicated (Jordan & Mitchell, 2015) that machine learning has become a popular technology for robot control and speech recognition. This highlights the massive growth of machine learning technology, which has been applied in different industries. Therefore, the integration of machine learning in this experiment is a good way to filter error messages more efficiently and provide richer response messages to help students complete their learning tasks.

Emotion detection: Programming courses are not easy to learn because they are specialized courses. Thus, when students encounter problems in learning, it is likely that they will have programming anxiety. Scott (2015) and Yildirim and Ozdener (2022) found that programming expertise requires deliberate practice but that students often experience programming anxiety at the beginning of their studies; they further found that programming anxiety affects students' practice behaviors and perceptions of learning performance and self-efficacy. Therefore, this experiment can be combined with machine learning algorithms to detect students' emotions. When students feel anxious about programming, the system will automatically provide appropriate hints or some related resources to help students practice continuously and complete their learning tasks.

6.2 Experimental Design 2: Note-taking Interactive E-learning Platform

This experiment recruited 59 students. However, the total number of students was not enough to divide the students into control and treatment groups. As such, it was challenging to use different groups to verify the experimental effects. Furthermore, this experiment was based on time constraints; we used a time span between the midterm and final exams. Therefore, the course design could be extended for a longer period of time in future work to collect more data for analysis. Regarding function, there are two future works in the study: (1) integrating social networks and (2) integrating distance education, as discussed below.

Integration of social networks: Recently, social networks have connected people worldwide, such as Instagram, Facebook, Line, etc. These networks have become an indispensable tool in daily life, and students typically prefer to use social networks to communicate. The present study indicates a similar point that social networks are continually influencing young people's lives; they use them for everything from entertainment to information. Therefore, this study incorporated social networks into the curriculum. This reveals a trend of social networks in educational activities that not only improves support, collaboration, and performance, but also has a positive impact on cognitive behaviors on social networks (Zachos et al., 2018).

Integrate distance education: In recent years, Covid-19 has seriously affected the entire world. Taiwanese students were suspended by Covid-19. Therefore, if this experiment can combine social network and distance education, it can help students not to be suspended and students can interact or discuss over the Internet. According to da Silva et al. (2021), distance learning has emerged as a new pedagogical approach to maintain continuity in education; teachers can analyze data to generate knowledge and behavioral data to adjust teaching strategies. Furthermore, distance learning systems can provide many different services such as recommendation systems and learning systems and personalized content, which can reduce the likelihood of students dropping out of school (da Silva et al., 2021).

References

Aleksić, V., & Ivanović, M. (2016). Introductory Programming Subject in European Higher Education. *Informatics in Education*, *15*(2).

Anderson, T. (Ed.). (2008). *The theory and practice of online learning*. Athabasca University Press.

Atinafu, B. (2021). Higher Education Students' Social Media Literacy in Ethiopia: A Case of Bahir Dar University. *Journal of Media Literacy Education*, 13(3), 86-96.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54.

Bau, D., Gray, J., Kelleher, C., Sheldon, J., & Turbak, F. (2017). Learnable programming: blocks and beyond. *Communications of the ACM*, *60*(6), 72-80.

Bell, T., Andreae, P., & Robins, A. (2014). A case study of the introduction of computer science in NZ schools. *ACM Transactions on Computing Education (TOCE)*, *14*(2), 1-31.

Blasco-Arcas, L., Buil, I., Hernández-Ortega, B., & Sese, F. J. (2013). Using clickers in class. The role of interactivity, active collaborative learning and engagement in learning performance. *Computers & Education*, *62*, 102-110.

Bohay, M., Blakely, D. P., Tamplin, A. K., & Radvansky, G. A. (2011). Note taking, review, memory, and comprehension. *The American journal of psychology*, *124*(1), 63-73.

Bolkan, S. (2015). Intellectually stimulating students' intrinsic motivation: The mediating influence of affective learning and student engagement. *Communication Reports*, *28*(2), 80-91.

Broadbent, J., & Lodge, J. (2021). Use of live chat in higher education to support self-regulated help seeking behaviours: a comparison of online and blended learner perspectives. *International Journal of Educational Technology in Higher Education*, *18*(1), 1-20.

Bui, D. C., Myerson, J., & Hale, S. (2013). Note-taking with computers: Exploring alternative strategies for improved recall. *Journal of Educational Psychology*, *105*(2), 299.

Bui, D. C., Myerson, J., & Hale, S. (2013). Note-taking with computers: Exploring alternative strategies for improved recall. *Journal of Educational Psychology*, *105*(2), 299.

Carretero, S., Vuorikari, R., & Punie, Y. (2017). The digital competence framework for citizens. *Publications Office of the European Union*.

Chang, W. C., & Ku, Y. M. (2015). The effects of note-taking skills instruction on elementary students' reading. *The Journal of Educational Research*, *108*(4), 278-291.

Chen, R. C., Chen, S. Y., Fan, J. Y., & Chen, Y. T. (2012). Grouping partners for cooperative learning using genetic algorithm and social network analysis. *Procedia Engineering*, *29*, 3888-3893.

Chien, Y. T., Lee, Y. H., Li, T. Y., & Chang, C. Y. (2015). Examining the effects of displaying clicker voting results on high school students' voting behaviors, discussion processes, and learning outcomes. *Eurasia Journal of Mathematics, Science and Technology Education*, *11*(5), 1089-1104.

Chu, Y. H., Hsu, S. M., Chang, C. C., & Hsu, S. H. (2020). An economy-wide assessment of artificial intelligence investment on manufacturing: a case study of Taiwan's semiconductor and ICT industries. *Modern Economy*, 11(5), 1040-1052.

Cooper, K. S. (2014). Eliciting engagement in the high school classroom: A mixed-methods examination of teaching practices. *American Educational Research Journal*, *51*(2), 363-402.

Cooper, S., Dann, W., & Pausch, R. (2000). Alice: a 3-D tool for introductory programming concepts. *Journal of computing sciences in colleges*, *15*(5), 107-116.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking-A guide for teachers.

Csizmadia, A., Standl, B., & Waite, J. (2019). Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education*, *18*(1), 41-67.

da Silva, L. M., Dias, L. P. S., Rigo, S., Barbosa, J. L. V., Leithardt, D. R., & Leithardt, V. R. Q. (2021). A literature review on intelligent services applied to distance learning. *Education Sciences*, *11*(11), 666.

Daniels, L. M., Goegan, L. D., & Parker, P. C. (2021). The impact of COVID-19 triggered changes to instruction and assessment on university students' self-reported motivation, engagement and perceptions. *Social Psychology of Education*, *24*(1), 299-318.

Dougherty, D. (2012). The maker movement. *Innovations: Technology, governance, globalization*, *7*(3), 11-14.

Falkner, K., Vivian, R., & Williams, S. A. (2018). An ecosystem approach to teacher professional development within computer science. *Computer Science Education*, *28*(4), 303-344.

Ferrari, A., & Punie, Y. (2013). DIGCOMP: A framework for developing and understanding digital competence in Europe.

Ferrer, J., Ringer, A., Saville, K., A Parris, M., & Kashi, K. (2020). Students' motivation and engagement in higher education: The importance of attitude to online learning. *Higher Education*, 1-22.

Fisher, D., & Frey, N. (2019). Peer Tutoring:"To Teach Is to Learn Twice". *Journal of Adolescent & Adult Literacy*, *62*(5), 583-586.

Fitriati, S. W. (2016). PSYCHOLOGICAL PROBLEMS FACED BY THE YEAR â€“ELEVEN STUDENTS OF MA NUHAD DEMAK IN SPEAKING ENGLISH. *English Education Journal*, *6*(1).

Fletcher, G. H., & Lu, J. J. (2009). Education Human computing skills: rethinking the K-12 experience. *Communications of the ACM*, *52*(2), 23-25.

Fletcher, G. H., & Lu, J. J. (2009). Education Human computing skills: rethinking the K-12

experience. *Communications of the ACM*, *52*(2), 23-25.
Fotaris, P., Mastoras, T., Leinfellner, R., & Rosunally, Y. (2016). Climbing up the Leaderboard: An Empirical Study of Applying Gamification Techniques to a Computer Programming Class. *Electronic Journal of e-learning*, *14*(2), 94-110.
Fraillon, J., Ainley, J., Schulz, W., Friedman, T., & Duckworth, D. (2020). *Preparing for life in a digital world: IEA international computer and information literacy study 2018 international report* (p. 297). Springer Nature.
Fraser, N. (2015, October). Ten things we've learned from Blockly. In *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)* (pp. 49-50). IEEE.
Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of educational research*, *74*(1), 59-109.
Gao, Q. (2021). Professional development and ICT literacy of college teachers based on FPGA and image target recognition education. *Microprocessors and Microsystems*, *80*, 103349.
García-Peñalvo, F. J., & Cruz-Benito, J. (2016, November). Computational thinking in pre-university education. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 13-17).
Giannakos, M. N., Pappas, I. O., Jaccheri, L., & Sampson, D. G. (2017). Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies*, *22*(5), 2365-2382.
Gnambs, T. (2021). The development of gender differences in information and communication technology (ICT) literacy in middle adolescence. *Computers in Human Behavior*, *114*, 106533.
Gomoll, A., Hmelo-Silver, C. E., Šabanović, S., & Francisco, M. (2016). Dragons, ladybugs, and softballs: Girls' STEM engagement with human-centered robotics. *Journal of Science Education and Technology*, *25*(6), 899-914.
Google (2015). Exploring Computational Thinking. Retrieved from https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview
Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends*, 60, 510-516.
Guo, Y. (2018). The Influence of Academic Autonomous Motivation on Learning Engagement and Life Satisfaction in Adolescents: The Mediating Role of Basic Psychological Needs Satisfaction. *Journal of Education and Learning*, *7*(4), 254-261.
Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE Bulletin*, *41*(1), 183-187.
Harlow, D. B., Dwyer, H. A., Hansen, A. K., Iveland, A. O., & Franklin, D. M. (2018). Ecological

Design-Based Research for Computer Science Education: Affordances and Effectivities for Elementary School Students. *Cognition and instruction*, *36*(3), 224-246.

Hayitov, U. H. (2020). ICT literacy of secondary school teachers. *European Journal of Research and Reflection in Educational Sciences*, *8*(9).

Hijriyah, E. M., Darmawan, E., & Zamzami, M. R. A. (2018). Enhancing Biology Students Motivation Through Classroom Action Research Based STAD Learning Model. *Indonesian Journal of Biology Education*, *1*(1), 9-16.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*, 263-279.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

Juškevičienė, A., & DagienĖ, V. (2018). Computational thinking relationship with digital competence. *Informatics in Education*, *17*(2), 265-284.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, *37*(2), 83-137.

Keller, J. M. (1983). Motivational design of instruction. In C. M. Regality (Ed.), Instructional design theories and models: An overview of their current status. *Hillsdale, NJ: Lawrence Erlbaum*. 384-434.

Keller, J. M. (2006). Development of two measures of learner motivation. *Unpublished Manuscript in progress. Florida State University*.

Khan, T., Johnston, K., & Ophoff, J. (2019). The impact of an augmented reality application on learning motivation of students. *Advances in Human-Computer Interaction*, *2019*.

Khan, T., Johnston, K., & Ophoff, J. (2019). The impact of an augmented reality application on learning motivation of students. *Advances in Human-Co*

Kim, H. S., Ahn, S. H., & Kim, C. M. (2019). A new ICT literacy test for elementary and middle school students in republic of Korea. *The Asia-Pacific Education Researcher*, *28*(3), 203-212.

Kim, H. S., Kil, H. J., & Shin, A. (2014). An analysis of variables affecting the ICT literacy level of Korean elementary school students. *Computers & Education*, *77*, 29-38.

Kim, Han Sung, et al. "Extending computational thinking into information and communication technology literacy measurement: Gender and grade issues." *ACM Transactions on Computing Education (TOCE)* 21.1 (2021): 1-25.

Kirschner, P., Strijbos, J. W., Kreijns, K., & Beers, P. J. (2004). Designing electronic collaborative learning environments. *Educational technology research and development*, *52*(3), 47-66.

Klašnja-Milićević, A., Vesin, B., & Ivanović, M. (2018). Social tagging strategy for enhancing e-learning experience. *Computers & Education*, *118*, 166-181.

Lai, C. F., Jeng, Y. L., & Huang, S. B. (2020). Improving programming skills: the use of learning style theory and the instant response supplement tool. *Library Hi Tech*.

Lau, W. W., & Yuen, A. H. (2014). Developing and validating of a perceived ICT literacy scale for junior secondary school students: Pedagogical and educational contributions. *Computers & Education*, *78*, 1-9.

Lee, S. J., & Chae, Y. G. (2012). Balancing participation and risks in children's internet use: The role of internet literacy and parental mediation. *Cyberpsychology, Behavior, and Social Networking*, *15*(5), 257-262.

Lin, P. Y., Chai, C. S., Jong, M. S. Y., Dai, Y., Guo, Y., & Qin, J. (2021). Modeling the structural relationship among primary students' motivation to learn artificial intelligence. *Computers and Education: Artificial Intelligence*, *2*, 100006.

Linder, S. P., Abbott, D., & Fromberger, M. J. (2006). An instructional scaffolding approach to teaching software design. *Journal of Computing Sciences in Colleges*, *21*(6), 238-250.

Margulieux, L. E., & Catrambone, R. (2021). Scaffolding problem solving with learners' own self explanations of subgoals. *Journal of Computing in Higher Education*, *33*, 499-523.

Marwan, S., Gao, G., Fisk, S., Price, T. W., & Barnes, T. (2020). Adaptive immediate feedback can improve novice programming engagement and intention to persist in computer science. In *Proceedings of the 2020 ACM conference on international computing education research* (pp. 194-203).

Nguyen, T. D., Cannata, M., & Miller, J. (2018). Understanding student behavioral engagement: Importance of student interaction with peers and teachers. *The Journal of Educational Research*, *111*(2), 163-174.

O'donnell, A. M., & King, A. (2014). *Cognitive perspectives on peer learning*. Routledge.

Pilkington, C. (2018). A playful approach to fostering motivation in a distance education computer programming course: Behaviour change and student perceptions. *The International Review of Research in Open and Distributed Learning*, *19*(3).

Pinto-Llorente, A. M., Martín, S. C., González, M. C., & García-Peñalvo, F. J. (2016). Developing computational thinking via the visual programming tool: Lego Education WeDo. In *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 45-50).

Price, M., Handley, K., Millar, J., & O'donovan, B. (2010). Feedback: all that effort, but what is the effect?. *Assessment & Evaluation in Higher Education*, *35*(3), 277-289.

Puspitarini, Y. D., & Hanif, M. (2019). Using Learning Media to Increase Learning Motivation in Elementary School. *Anatolian Journal of Education*, *4*(2), 53-60.

Puspitarini, Y. D., & Hanif, M. (2019). Using Learning Media to Increase Learning Motivation in Elementary School. *Anatolian Journal of Education*, *4*(2), 53-60.

Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., & Ab Rahman, R. (2012). Major problems in basic programming that influence student performance. *Procedia-Social and Behavioral Sciences*, *59*, 287-296.

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, *25*(1), 54-67.

Ryan, R. M., & Deci, E. L. (2020). Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions. *Contemporary educational psychology*, *61*, 101860.

Sadera, W. A., Robertson, J., Song, L., & Midon, M. N. (2009). The role of community in online learning success. *Journal of Online Learning and Teaching*, *5*(2), 277-284.

Salame, I. I., & Thompson, A. (2020). Students' Views on Strategic Note-Taking and Its Impact on Performance, Achievement, and Learning. *International Journal of Instruction*, *13*(2), 1-16.

Santos, G. M., Ramos, E. M., Escola, J., & Reis, M. J. (2019). ICT Literacy and School Performance. *Turkish Online Journal of Educational Technology-TOJET*, *18*(2), 19-39.

Scherer, R., & Siddiq, F. (2019). The relation between students' socioeconomic status and ICT literacy: Findings from a meta-analysis. *Computers & Education*, *138*, 13-32.

Scott, M. (2015). *Self-Beliefs in the Introductory Programming Lab and Games-based Fantasy Role-Play* (Doctoral dissertation, Brunel University).

Shi, Y., Tong, M., & Long, T. (2021). Investigating relationships among blended synchronous learning environments, students' motivation, and cognitive engagement: A mixed methods study. *Computers & Education*, *168*, 104193.

Shih, T. K., Gunarathne, W. K. T. M., Ochirbat, A., & Su, H. M. (2018). Grouping Peers Based on Complementary Degree and Social Relationship using Genetic Algorithm. *ACM Transactions on Internet Technology (TOIT)*, *19*(1), 2.

Shyr, W. J., Hsieh, Y. M., & Chen, C. H. (2021). The Effects of Peer-Based Instant Response System to Promote Learning Performance, Intrinsic Motivation and Self-Efficacy. *Sustainability*, *13*(8), 4320.

Sinha, S., Rogat, T. K., Adams-Wiggins, K. R., & Hmelo-Silver, C. E. (2015). Collaborative group engagement in a computer-supported inquiry learning environment. *International Journal of Computer-Supported Collaborative Learning*, *10*(3), 273-307.

Smith, J., Guimond, F. A., Bergeron, J., St-Amand, J., Fitzpatrick, C., & Gagnon, M. (2021). Changes in students' achievement motivation in the context of the COVID-19 pandemic: A function of extraversion/introversion?. *Education Sciences*, *11*(1), 30.

Starr-Glass, D. (2020). Encouraging engagement: video-conference augmentation of online distance learning environments. *On the Horizon*.

Su, C. Y., & Chen, C. H. (2018). Investigating the effects of flipped learning, student question generation, and instant response technologies on students' learning motivation, attitudes, and engagement: A structural equation modeling. *EURASIA Journal of Mathematics, Science and Technology Education*, *14*(6), 2453-2466.

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

Witherby, A. E., & Tauber, S. K. (2019). The current status of students' note-taking: Why and how do students take notes?. *Journal of Applied Research in Memory and Cognition*, 8(2), 139-153.

Wu, H., Li, S., Zheng, J., & Guo, J. (2020). Medical students' motivation and academic performance: the mediating roles of self-efficacy and learning engagement. *Medical education online*, *25*(1), 1742964.

Wulandari, N. A., Amalia, S., & Ramdhani, M. I. (2022). Investigating speaking difficulties of senior high school students: linguistics and psychological problems. *ENGLISH FRANCA: Academic Journal of English Language and Education*, *6*(1), 151-166.

YILDIRIM, O. G., & OZDENER, N. (2022). The Development and Validation of the Programming Anxiety Scale. *International Journal of Computer Science Education in Schools*, *5*(3), 17-34.

Yıldırım, T., & Canpolat, N. (2019). An investigation of the effectiveness of the peer instruction method on teaching about solutions at the high-school level. *Egitim ve Bilim*, 44(199).

Yuan-Tai-Ke. (2021). National Science and Technology Development Plan. Extracts from https://www.nstc.gov.tw/nstc/attachments/fd10ecfa-f319-44fe-8058-f379d3de22c2.

Yuliana, I., Hermawan, H. D., Prayitno, H. J., Ratih, K., Adhantoro, M. S., Hidayati, H., & Ibrahim, M. H. (2021). Computational Thinking Lesson in Improving Digital Literacy for Rural Area Children via CS Unplugged. In Journal of Physics: Conference Series (Vol. 1720, No. 1, p. 012009). IOP Publishing.

Zachos, G., Paraskevopoulou-Kollia, E. A., & Anagnostopoulos, I. (2018). Social media use in higher education: A review. *Education Sciences*, *8*(4), 194.

Zeinstra, L., Kupers, E., Loopers, J., & de Boer, A. (2023). Real-time teacher-student interactions: The dynamic interplay between need supportive teaching and student engagement over the course of one school year. *Teaching and Teacher Education*, *121*, 103906.

Zou, T. X., Ko, E. I., & Mickleborough, N. (2012). Promoting multi-layered peer learning in a course on engineering grand challenges. *Procedia-Social and Behavioral Sciences*, *56*, 74-87.