Available online at
http://journale-academiauitmt.uitm.edu.my
e-Academia Journal

e-Academia Journal of UiTM Cawangan Terengganu 12 (2) 224-239, November 2023

# Innovation of Digital Multimedia: Recognizing the Advantages

**Azyan Yusra Kapi@Kahbi[1], *Azrina Suhaimi[1], Harshida Hasmy[1] & Mohamad Faizal Ab Jabal[1]**

[1]Department Computing Sciences Studies, College of Computing, Informatics and Mathematics,
Universiti Teknologi MARA, Johor Branch, Pasir Gudang Campus,
81700 Johor, Malaysia

*Corresponding author's email: azrin253@uitm.edu.my

Submission date: 12 September 2023 Accepted date: 16 November 2023 Published date: 26 November 2023

*To cite this article:* Kapi@Kahbi, A. Y., Suhaimi, A., Hasmy, H. & Jabal, M. F. A. (2023). Innovation of Digital Multimedia: Recognizing the Advantages. *e-Academia Journal of UiTM Cawangan Terengganu, 12*(2), 224-239

## ABSTRACT

This study explores the effectiveness of an innovative multimedia application as a teaching tool for improving students' understanding of C++ functions. The intervention incorporated advanced visualization techniques and rich multimedia elements to create a dynamic learning experience. The primary objective was to assess the intervention's impact on students' comprehension and evaluate its educational significance. A total of 33 participants underwent pre-test and post-test assessments, with the normalized gain (N-Gain) serving as the key metric for improvement. The calculated N-Gain was approximately 0.6336, indicating a moderate improvement in students' performance following the intervention. The discussion interprets the N-Gain in the context of the study's objectives and practical implications. Notably, the study underscores the e-content application's effectiveness in achieving educational goals and its potential for broader implementation in computer science education. Despite limitations such as sample size and context-specific factors, this research highlights the importance of innovative educational approaches in the digital age. It contributes to the evolving landscape of educational technology and pedagogy, offering valuable insights for educators aiming to create engaging and impactful learning experiences in computer science and related fields.

***Keywords:*** *C++ functions, normalized gain, educational technology, multimedia, computer science education*

## 1.0 INTRODUCTION

In the past few years, C++ has often been perceived as one of the most challenging and demanding courses for university students. Beginners in programming may find these courses difficult due to their initial lack of proficiency in the subject (Ali & T. Smith, 2014). They may become frustrated about their inability to understand the programming topics, which will negatively affect their performance on the ongoing assessments (Rahmat et al., 2012). At Universiti Teknologi MARA Pasir Gudang Campus, the curriculum for diploma students in mechanical and civil engineering mandates the inclusion of C++ programming in the syllabus for Part 2. The results of the latest assessment indicate that the students tended to overlook and demonstrate incomplete comprehension of questions about the topic of C++ functions. In light of this matter, an initial investigation has been carried out to identify potential solutions.

Numerous approaches, as documented in the literature, have been utilised to facilitate students in developing an affinity for studying C++ programming and improving their understanding and visualisation of fundamental principles. Multimedia technologies could be an effective option in the current digital learning environment for improving students' understanding of programming languages (Ramli et al., 2015). This study introduces a novel e-content application that aims to enhance the understanding of the C++ Function topic through effective visualisation techniques, including the utilisation of rich multimedia elements. Additionally, the program incorporates the use of created analogies and direct practice with questions and answers to further facilitate comprehension. The platform provides concise visual illustrations and enables students to engage in interactive quizzes, receiving immediate feedback and outcomes. It also provides convenient accessibility across many devices, including desktop computers, laptops, and mobile phones.

In the contemporary era of digital education, the integration of multimedia resources such as iSpring Presenter has the potential to significantly enhance students' comprehension of programming languages. This study is based on the ADDIE development model, which has five steps: analysis, design, development, implementation, and evaluation. The goal is to improve the ability of multimedia learning tools to get students interested in programming (Anwar et al., 2012). However, there is a lack of prior research in the development of interactive multimedia for programming education that integrates analogy and practice using iSpring Presenter as a means to enhance students' engagement in the learning process.

## 2.0 LITERATURE REVIEW

In this section, a thorough examination of the critical discussions and insights derived from a wide variety of sources is provided, shedding light on the current state of knowledge and trends in the domain. Throughout the literature review, various aspects of programming education are explored, including theoretical frameworks and teaching tools.

### 2.1 Theoretical Background

The theoretical foundation of any educational research study is essential to shaping its design, objectives, and interpretation. In this section, three prominent theories underpinning programming education are examined: Cognitive Load Theory (CLT), Constructivist Learning Theory, and Dual Coding Theory. Understanding the complexities of programming education is possible through the use of these theories, which provide valuable insights into how learners acquire and process knowledge.

#### *2.1.1 Cognitive Load Theory (CLT)*

In the context of instructional design and learning, Cognitive Load Theory (CLT) has emerged as an important theoretical framework (Van Merrienboer & Sweller, 2005). To achieve optimal learning outcomes, it emphasizes the importance of managing cognitive load, which represents the number of cognitive resources learners allocate to process information (Haryana et al., 2022). While learning programming languages and solving issues involves significant cognitive demands, CLT has been shown in the literature to be relevant to programming education (Berssanette & de Francisco, 2021). CLT emphasizes that instructional methods should align with cognitive processes to optimize learning. Therefore, it is imperative to understand how to minimize non-essential cognitive load while maximizing essential cognitive load. Additionally, according to Renkl et al. (2009), providing students with too much complex, abstract material can hinder their learning in a programming context. Several studies have demonstrated that well-structured programming exercises and scaffolded learning activities reduce cognitive overload, allowing learners to gain a deeper understanding of programming concepts such as studies by Salleh et al. (2018) and Guzman et al., (2019).

#### *2.1.2 Constructivist Learning Theory*

Constructivism posits that learning is an active and constructive process where learners build their understanding through interaction with their environment and social interactions with others (Narayan et al., 2013). In the context of programming and understanding abstract concepts like C++ functions, tools that allow students to relate newly acquired information to prior knowledge are highly beneficial.

According to the literature, programming is a discipline that embodies the constructivist approach to problem-solving (Wulf, 2005). Students in programming courses often engage in hands-on activities, address real-world coding challenges, and collaborate with peers, mirroring the principles of constructivism. The discussion focuses on how constructivist methods, such as project-based learning and collaborative coding exercises, can enhance programming education by encouraging active engagement and the development of knowledge. Furthermore, it emphasizes the need to foster problem-solving skills and critical thinking among programming students by creating a supportive and interactive learning environment.

#### *2.1.3 Dual Coding Theory*

According to the Dual Coding Theory by Paivio (1991), humans have two distinct but interconnected systems for processing information: a verbal system for language, and a nonverbal system for visual and spatial perception. It is evident from Dual Coding Theory's relevance to programming education that the process of programming consists of both verbal elements, such as the syntax and explanation of algorithms, and non-verbal elements, such as visual representations, flowcharts, and diagrams (C.-H. Wang & Wu, 2022). Understanding how learners can enhance their understanding of programming concepts by leveraging both cognitive systems is very crucial. Thus, a strong emphasis is placed on the potential benefits of incorporating visual and verbal elements into the teaching of programming. Visual aids along with textual explanations can assist in facilitating a deeper and more comprehensive understanding of programming concepts, according to this study (L. Wang & Li, 2019). Moreover, it underscores the necessity of considering learners' dual coding capabilities when designing instructional materials and activities.

### 2.2 Related Works on Teaching Tools for Programming

This section provides a comprehensive assessment of the existing teaching tools and approaches utilized in programming education. The following sections will provide an overview of related works on teaching tools for programming.

#### *2.2.1 Visualization*

The integration of visualization techniques in programming education has shown promise in enhancing students' understanding of complex programming concepts. Visual representations can help learners grasp abstract ideas, data structures, and algorithms more effectively. The work by Ramli et al., (2015) indicates that the visualization approach in teaching subtopic arrays in programming was successfully employed to improve the learning experience. In addition, Sorva et al., (2013) have pointed out that a review of several studies that use visualizations to teach introductory programming has shown positive results. When working with complex information structures, programming can be extremely complicated. Augmented Reality allows for the visualisation of real-time data processing methods and it appears to be a highly promising technological advancement for teaching programming (Theodoropoulos & Lepouras, 2021). Furthermore, the discussion surrounding the use of visualization in programming education underscores the need for instructors to carefully select and design visual aids that align with their pedagogical goals and the diverse learning styles of their students.

#### *2.2.2 Multimedia in Learning*

Within the field of education, the use of multimedia components in teaching and learning has become a prevalent and widely adopted approach. The utilisation of multimedia, which includes many forms such as images, text, animations, music, and visualisations, has proven to be a highly effective means of improving understanding in particular academic disciplines (Kapi et al., 2017; Shunkov et al., 2022). An illustrative instance of incorporating multimedia in programming education is presented by Bottino et al. (2023). The authors suggest a pedagogical approach that employs a multimedia programming environment to teach programming in primary schools. This approach utilises storytelling as a means to enhance computational fluency among both students and teachers. By engaging in the creation of digital stories, learners are encouraged to comprehend and effectively utilise computer code in conjunction with linguistic structures.

Additionally, while multimedia can enhance engagement, there is a need to strike a balance to prevent overwhelming students or distracting them from core learning objectives (Berk, 2009). Further examination of the discussion regarding multimedia in programming education reveals the importance of effective implementation and integration. Instructors should be equipped with the knowledge and skills to create and incorporate multimedia resources that align with their teaching objectives. Moreover, research should continue to explore the impact of multimedia on learning outcomes and investigate strategies for optimizing its use in programming instruction.

#### *2.2.3 Interactive Engagement, Quizzes and Gamification*

Interactive engagement strategies, quizzes, and gamification have emerged as potent tools for enhancing programming education. Studies suggest that these approaches can foster a more enjoyable and immersive learning experience, which can be particularly beneficial in the context of programming education. Gamification elements, such as coding challenges and point-based systems, and leader boards make the learning process more engaging and competitive (Yunus & Hua, 2021). Similarly, quizzes which provide real-time feedback also promote student engagement and motivation (Ibrahim et al., 2018). The research conducted by Swacha et al. (2023) demonstrates a beneficial influence on the educational achievements observed upon completion of a programming course, specifically when a gamified Massive Open Online Course (MOOC) is employed as a platform for learning Python programming.

However, it is crucial to acknowledge that not all interactive and gamification techniques are equally effective. Their success depends on various factors, including the alignment with learning objectives, the design of the activities, and the level of student engagement (Korn, 2022). Careful consideration and adaptation are required to ensure these strategies enhance, rather than distract from, the learning experience (Bekk et al., 2022). In moving forward, instructors should continue to explore innovative ways to incorporate interactive engagement and gamification into programming courses. Research should also focus on identifying the optimal balance between traditional instruction and these interactive approaches, as well as assessing their impact on learning outcomes.

#### *2.2.4 Analogies*

Analogies are comparisons between unfamiliar concepts or ideas and familiar ones, designed to simplify understanding. In the context of programming education, analogies have been employed to bridge the gap between abstract coding concepts and real-world experiences. Jiménez Toledo et al., (2021) suggest that well-crafted analogies can enhance learners' comprehension and retention of programming concepts. A study by Dunican (2002) demonstrated that analogies can significantly improve students' ability to understand and apply programming constructs, such as loops and conditionals. While the use of analogies is widely recognized in fields like physics and biology to simplify complex concepts, there is a relative scarcity of literature that systematically investigates their effectiveness in computer science, especially in topics like functions in C++.

### 2.3 Related Works On E-Learning Authoring Tool

During the last decade, multimedia content has grown substantially in importance in tertiary and higher education. The emergence and growth of e-learning in tertiary and higher education has necessitated the development of sophisticated authoring tools. One prominent trend is the increasing emphasis on interactive and technology-driven tools based on e-learning authoring tools. This review delves into the advancements and features of three predominant tools: Adobe Captivate, Articulate Storyline, and iSpring. These tools enable educators to create engaging, interactive, and multimedia-rich content tailored for online platforms.

Adobe Captivate stands out for its advanced functionalities, such as VR capabilities, responsive design, and software simulations. Several studies have used Adobe Captivate to develop their e-learning materials which consist of digital multimedia, such as Boukhechba and Bouhania (2019) and Alabay and Bastürk (2021).

Articulate Storyline is renowned for its intuitive interface and powerful features, allowing for the creation of complex interactions, animations, and branching scenarios (Hadza et al., 2020). Its capability to mimic software tasks makes it particularly useful for software training. Versatility in Articulate Storyline can be seen in its adaptability across devices, ensuring that learners have consistent experiences whether they access content on a desktop, tablet, or mobile (Linn & Rea, 2016).

iSpring Suite has emerged as a comprehensive e-learning authoring toolkit that seamlessly integrates with PowerPoint. Its user-friendliness is often highlighted, making it a preferred choice for educators transitioning from traditional to online teaching (Amali et al., 2019). The tool supports the creation of quizzes, interactive simulations, and screencasts. Furthermore, according to Vikulova et al. (2018), many reports have demonstrated that the implementation of iSpring in language classes in Moscow has resulted in positive outcomes when compared to conventional teaching approaches. The commendable level of student engagement has effectively mitigated the challenges associated with developing modules using iSpring. The modules developed by iSpring effectively facilitate teaching methods, with a specific emphasis on interactive ways that enable students to independently regulate the speed at which knowledge is delivered, thereby accommodating their individual capabilities. The user-friendly design of iSpring facilitates the process of module creation for instructors.

In the realm of scientific disciplines, particularly within the captivating field of chemistry (Hanika & Guspatni, 2023), the use of iSpring has been demonstrated to be highly valuable. For subjects like stoichiometry, renowned for their complexity, iSpring has elucidated intricate chemical concepts, laws, and formulas. The implementation of the prompting question strategy is advantageous for students, as it leads to enhanced comprehension, knowledge creation, and concept assimilation. Romisa (2023) established iSpring as a significant tool in the field of computer science education, distinguishing her approach by implementing it specifically on the Android platform. This strategic choice fosters exploration and active student participation. The open-source nature of Android invites contributions from diverse perspectives, enhancing the learning experience. Students engage more actively and contribute their ideas, enriching discussions.

Overall, iSpring has proven its worth in education. It elevates comprehension, engagement, and interest, allowing students to actively contribute to discussions. Educators play a crucial role in harnessing technology effectively and ensuring successful knowledge transfer and understanding. In this evolving educational landscape, iSpring stands as a valuable ally for both educators and learners and it has been chosen as a tool to create digital multimedia for teaching C++ functions in this study.

In addition to considering the technological tools employed as teaching aids, it is imperative to assess the efficacy of teaching strategies. There are several methodologies available for assessing the effectiveness of teaching strategies and interventions. The common approach to testing students is to administer a pre-test and a post-test. Using these tests, researchers can evaluate the effect of a specific instructional intervention on student learning outcomes before and after the intervention. Researchers can gain valuable insight into the effectiveness of different educational methods, tools, or approaches by comparing the results of pre-tests and post-tests as summarized in Table 1.

**Table 1: List of Research Studies that Used Pre-Post Tests to Measure the Performance of Students**

| Reference(s) | Descriptions | Topic/Subject | Type of Test | Result |
|---|---|---|---|---|
| **(Neureiter et al., 2020)** | The online gaming platform Kahoot! was anonymously implemented before and after “classical” medical education which included discussions of histological slides for each tumor entity using Microsoft PowerPoint-based presentations in combination with microscopical demonstrations. | (Histo-) pathology in medical | Mann–Whitney U-test or Student´s t-test | The findings indicate that it is feasible and effective to increase students' engagement and motivation for active participation during pathology instruction in the classroom. |
| **(Anwar et al., 2019)** | The objective is to create a multimedia learning program with iSpring Presenter in order to enhance students' engagement and motivation in the field of Mathematics. | Mathematics | N-Gain test | The N-Gain value of 0.704 indicates that interactive mathematics multimedia learning using iSpring Presenter was highly effective in increasing students' interest in Mathematics. |
| **(Yunus & Hua, 2021)** | This study explores Quizizz, a gamified educational tool in | English (irregular verbs) | Mann-Whitney U test | The outcomes of this study suggest that the utilization of game- |

| | | | | |
|---|---|---|---|---|
| | delivering irregular English verbs. | | | based learning such as Quizizz in language instruction is advantageous, as it effectively enhances students' engagement and acquisition of English irregular past verbs. |

Table 1 summarizes three different educational studies and their findings related to enhancing student engagement and motivation using various digital tools and approaches. In research by Neureiter et al., conducted a study anonymously implementing the online gaming platform Kahoot! before and after traditional medical education sessions regarding histopathology in 2020. Students were engaged and motivated during pathology instruction when they were given the Mann-Whitney U-test, which is also known as the Student's t-test. The author (s) must be able to demonstrate a critical understanding of relevant theories/concepts which underpin the research work.

Another study by Anwar et al., published in 2019, aimed to increase student engagement and motivation in mathematics using iSpring Presenter. Using the N-Gain test, they evaluated the effectiveness of their approach and found a high N-Gain value of 0.704, indicating that interactive mathematics multimedia learning using iSpring Presenter greatly increased students' interest in mathematics.

Using Quizizz, a gamified educational tool, Yunus & Hua (2021) conducted a study that focused on the teaching of irregular English verbs using Quizizz. To examine the efficacy of this game-based approach, the researchers used the Mann-Whitney U test and found that it greatly enhanced students' engagement and ability to acquire irregular past verbs in English. As a result of these studies, it can be concluded that digital tools and gamification have the potential to improve student engagement and motivation across a wide spectrum of educational subjects.

In conclusion, the theoretical background section of the literature review sets the stage for the subsequent sections of the study. It elucidates the significance of Cognitive Load Theory, Constructivist Learning Theory, and Dual Coding Theory in the context of programming education and provides the necessary theoretical lenses. The reviewed studies demonstrate the substantial positive impact of digital tools and multimedia platforms like iSpring, Kahoot!, and Quizizz on student engagement and motivation across various subjects. These findings underscore the importance of integrating technology into education to enhance learning experiences, boost student interest, and improve overall comprehension. As education continues to evolve, educators should embrace innovative tools and strategies to foster active participation and knowledge acquisition among students, ensuring their readiness for the challenges of the 21st century. Innovative teaching methodologies, rooted in sound educational theories, have the potential to revolutionize how programming concepts like functions in C++ are taught. While individual tools and methods exist, a comprehensive tool that integrates visualization, multimedia, analogy, and quizzing remains elusive. Combined with rigorous efficacy evaluations, such a tool can bridge the existing divide and substantially raise programming instruction standards.

## 3.0 METHODOLOGY

The basis of our investigation is founded upon the ADDIE concept. The ADDIE model is a structured instructional design framework that is employed to facilitate the development of impactful educational experiences. The teaching tool was created using the ADDIE approach, which is a widely recognised instructional design model. The ADDIE approach consists of five distinct stages, namely Analysis, Design, Development, Implementation, and Evaluation. The following subsection describes the five phases of ADDIE.

### 3.1 Analysis

Several main procedures are carried out during this stage which are (i) problem identification, (ii) subtopics identification, and (iii) technical considerations. The details are presented in the next subsection:

#### *3.1.1 Problem Identification*

In order to identify the current knowledge level of the students, a sample of 69 students' test results are analysed to identify areas of strength and weakness, which can indicate where instruction is most required or how current instructional methods may be adapted for improved outcomes. The results of the analysis are shown in Figure 1.

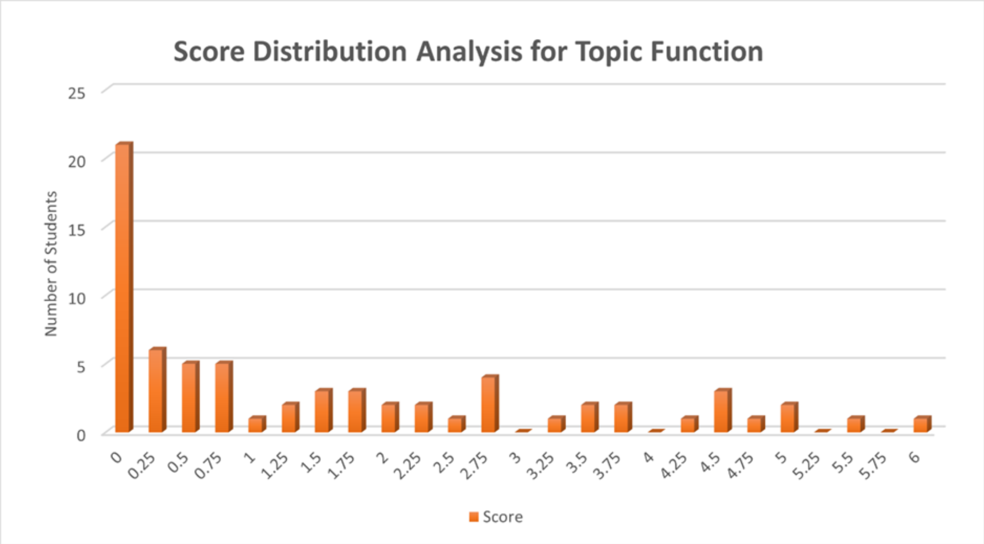


**Figure 1: Score Distribution Analysis for Topic Function in C++**

Figure 1 shows the variability in the number of students receiving different marks, which can be indicative of a diverse range of student performance. The analysis indicates that out of 69 students, 21 received the lowest mark of 0, while only 1 student received the highest mark of 6. Comparing the results with other topics in the C++ programming course, it can be concluded that students find it hard to understand specific concepts of Functions due to their complexity.

#### *3.1.2 Subtopics Identification*

The test results can be used to identify what students already know and where gaps exist in order to design which subtopics of functions should be included in the teaching tools, as well as how analogies would be most effective in which areas.

#### *3.1.3 Technical Considerations*

Other than that, it has been observed that most lecturers use a dry PowerPoint presentation and practice questions during classes, which does not assist students in understanding the theoretical concept of functions. Furthermore, it is essential to consider the technical resources available to students, such as whether they will be accessing the course through a desktop, tablet, or mobile device. Therefore, by the existing literature, the teaching tool introduced an innovation using a combination of three elements: analogy, multimedia, and practice quizzes, intending to improve student performance in the selected topic of Function in C++.

### 3.2 Design

A design stage consists of three important components: the design of an analogy (story), digital multimedia elements, and questions related to practices. It takes about two weeks for the entire design process to be completed. The subsequent sections provide a comprehensive overview of the design process.

#### *3.2.1 Design and Structure of the Teaching Tool, including Analogies and Subtopics*

An example of a wedding planner is used in the teaching tool as a means of illustrating how the function is applied in real-life situations. In the beginning of the teaching tools, analogies of the wedding planner are presented, and an explanation emerges based on this case.

The design of codes also incorporates syntax in addition to analogies. Once the student has developed a comprehensive comprehension of the concept of function, they are then exposed to and directed towards syntax. Moreover, this particular issue faced by students poses a challenge in establishing a connection between the fundamental principles and the syntactical aspects. The content of the teaching tool was organised and constructed as follows (numbers are used to indicate the outline of the tools):

- **Pre-Test -** Before using the teaching tool, it is necessary to conduct a pre-assessment to determine the level of understanding of the students.
- **(1.0) Analogy of Function in Real-Life Situations -** Introduction to the concept of functions through the establishment of analogies with real-life situations. This aids students in establishing a connection with the concept of functions and comprehending their significance in the context of problem-solving.
- **(1.1) Type of User-Defined Function -** A discussion of the various types of user-defined functions. As opposed to built-in functions that are part of a programming language, these are functions that programmers create to accomplish specific tasks.
- **(1.2) Summary -** An overview of the points discussed under analogy and user-defined functions.
- **(2.0) Definition of a Function -** An introduction to the concept of function in the context of programming is provided.
- **(2.1) Terminology -** Detailed explanations of the function terms, such as function call and function definition.
- **(2.1.1) Function Call -** A function call describes how and when a function is invoked or executed. **(2.1.2) Function Prototype/Declaration -** In the form of a prototype/declaration, a function is defined by its name, return type, and parameters, but not by its body.
- **(2.1.3) Function Definition (Header/Body) -** The definition of the function (header/body) consists of the actual code about that function. The header contains the return type, the name of the function, and the parameters, while the function's body contains the set of instructions that will be executed by the function.
- **(2.2) Type of Parameter -** This lesson examines the types of parameters that can be passed to a function, including formal and actual parameters.
- **(2.3) Summary -** It is a summary of the function's definition and associated terminology.
- **(3.0) Practice Quiz -** Students are required to undertake a brief assessment to evaluate their comprehension of the material they have thus far studied.
- **(4.0) Example of Function -** An example of a function is designed to provide concrete examples that further explain the concept of functions.
- **(4.1) How are Values Passed between Functions? -** The process of exchanging values across functions involves the following components:
- **(4.1.1) Through parameter passing by value**
- **(4.1.2) Through return keyword**
- **(4.1.3) Through parameter passing by reference**
- **(4.2) Summary -** The purpose of this section is to provide a concise explanation of how functions work through examples.
- **(5.0) Practice Quiz -** There will be one more quiz, this time focusing on the more advanced topics discussed in the examples.
- **Post-Test -** The purpose of the final assessment is to determine whether the material presented in the teaching tool has been understood and retained by the student.

### *3.2.2 Integration of Digital Multimedia Elements*

In the modern digital landscape, digital multimedia elements have become increasingly important. A multimedia element creates an engaging experience by combining text, images, video, music, and animations (Pavithra et al., 2018). Each of these components serves its unique purpose:

- **Text**: Establishes the foundation and context for the discussion. In most multimedia presentations, text serves as the foundation, providing details and explanations that may be supported or enhanced by other elements in the teaching tool.
- **Images**: The use of images will enhance engagement and facilitate understanding of the content. It is often faster to convey an idea through an image than to read text.
- **Video:** Provides a dynamic method of demonstrating concepts, telling stories, or providing detailed explanations in the teaching tool. Incorporating motion and sound, it can capture the attention of the audience and make complex ideas more understandable.

- **Animations:** An animation is a video representation that is stylized or abstract, similar to a video. An animation can simplify complicated processes in a teaching tool or provide a fresh, lively feel to its content.

PowerPoint allows users to create storyboards for multimedia presentations. In this context, a storyboard is comparable to a visual script. It describes one slide or scene at a time and how the content will emerge. To assist in visualizing the design concept for the teaching tool, a storyboard is produced as shown in Figure 2.

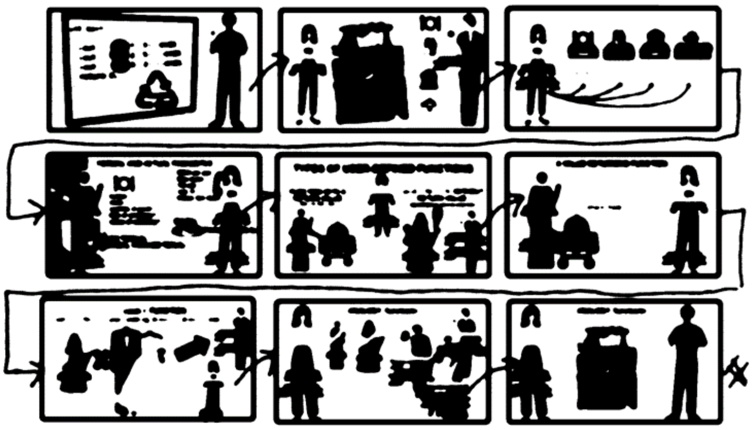

**Figure 2: A Snippet from the Design Storyboard Teaching Tool**

Figure 2 illustrates a snippet of the design storyboard that serves as a guide for the development of the teaching tool. When implemented with proficiency, interactive multimedia components have the potential to enhance a basic presentation, resulting in an engaging and enlightening experience for its viewers.

***3.2.3 Practice Quizzes and Pre-Post Tests Design***

Through practice quizzes, students can reinforce their learning by recalling and retrieving information. Furthermore, it provides real-time feedback, allowing students to identify areas where they are strong and weak. Besides strengthening memory retention, it can also serve as spaced repetition. The design for practice quizzes considers several aspects such as:

- Relevance: Questions must relate directly to the learning material.
- Variety: Use a variety of question types, such as multiple choice, true/false, short answer questions, and matching questions.
- Feedback: Provide explanations both for correct and incorrect answers.
- Difficulty: In an ideal scenario, there should be a range of difficulty levels in the quiz.
- Frequency: Conduct quizzes regularly to reinforce learning.

This research will also use a single-group pre-test and post-test design in addition to the practice quizzes. A similar group of students will be assessed before and after exposure to the teaching tool for C++ Functions. Pre- and post-test questions were designed taking several aspects into account, as previously described in the design of practice quizzes.

**3.3 Development**

The development of teaching tools encompasses two fundamental procedures that follow the design from the previous stage. These procedures are described in detail in the following sections.

***3.3.1 Creation of the Teaching Tool and Assessment***

Development of the teaching tool and its accompanying assessment requires the integration of various software applications that cater to different aspects of the delivery and evaluation of educational content. As outlined in Table 2, there are five main application software used in the development of the teaching tool and the assessment as depicted in Table 2.

**Table 2: List of Application Software used in the Development of Teaching Tool and Assessment**

| No. | Application Software | Purpose |
|---|---|---|
| **1** | Microsoft PowerPoint with iSpring add-in | To combine multiple media elements such as animation, sounds, text, graphics, and video. |
| **2** | iSpring Cam Pro | To record screen and to edit video. |
| **3** | Wondershare Filmora | To edit video and voice over. |
| **4** | iSpring QuizMaker | This tool is designed for creating interactive quizzes and surveys. |
| **5** | Microsoft Form | Platform to distribute pre and post-test to respondent. |

Table 2 provides a list of the software applications that were used in the development process and their purposes. Figure 2 depicts the process by which analogies were generated by the combined utilisation of animation and auditory elements, resembling the techniques employed by a narrator. Most of the animation was constructed using Microsoft PowerPoint with embedded iSpring, as presented in Figure 2.

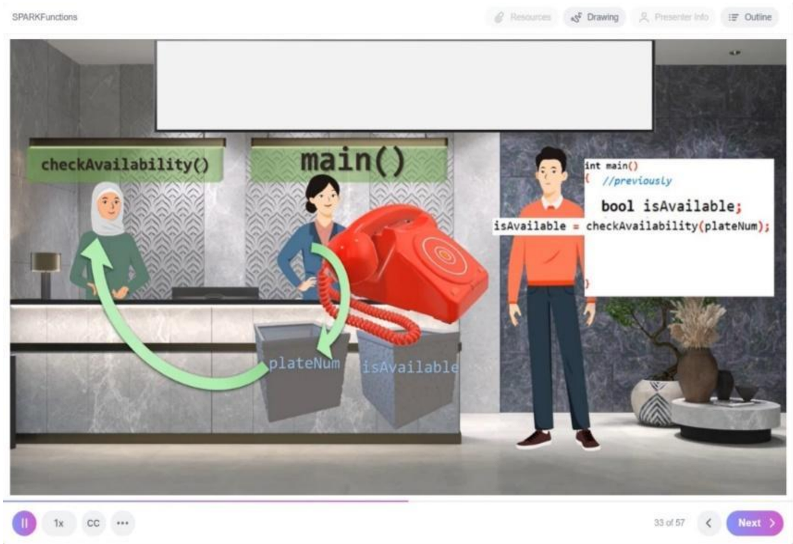


**Figure 2: Screenshot that Represents Multimedia for Mimicking Function Calls in C++**

In the teaching tool, screen recordings were made with iSpring Cam Pro, and practice quizzes were developed with iSpring QuizMaker. An example of a practice quiz for C++ functions can be seen in the screenshot from the teaching tool, as portrayed in Figure 3.

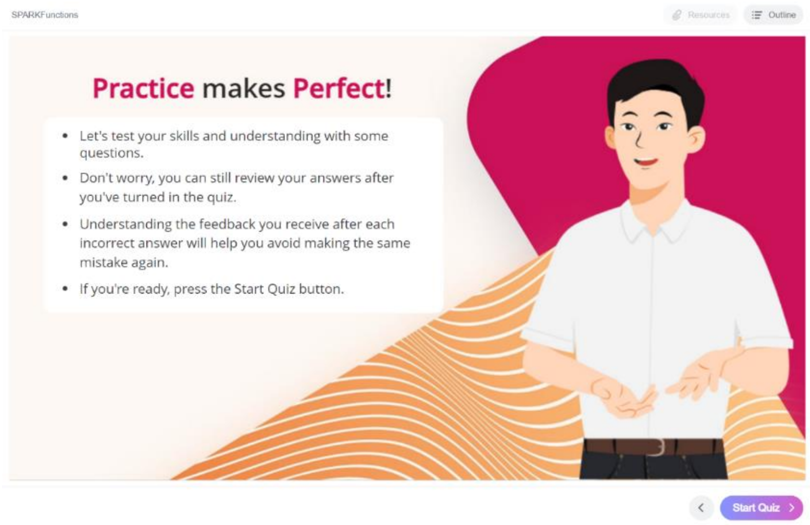


**Figure 3: Screenshot of a Practice Quiz for C++ Functions taken from the Teaching Tool**

While iSpring does include the capability to generate surveys, Microsoft Form, as depicted in Figure 4, was chosen as the preferable tool for constructing the pre- and post-test questions due to its ability to collect respondents' scores in a single spreadsheet, hence facilitating data analysis. The challenge is in the limited functionality of iSpring QuizMaker, as it solely offers score reporting via individual email notifications, hence posing difficulties in the analysis and management of user responses. Furthermore, a significant portion of the visuals utilised in this work were sourced from the iSpring Content Library. It is important to note that proper credit and acknowledgement have been given to the creators of the PowerPoint templates.

#### *3.3.2 Review and Testing the Prototype*

It is important to review and test a prototype during the development process to ensure that the prototype complies with the objectives. After its development, the teaching tool is subsequently evaluated by two lecturers who possess as subject matter experts. As a result, the teaching tool has been corrected as a result of their valuable input.

In conclusion, the integration of these software applications enables a holistic approach to the creation of content and the assessment of the teaching tool. The services they provide collectively encompass many aspects of the educational process, ranging from the presentation of content and the creation of videos to the collection. Through this synergy, an effective and engaging teaching tool with robust assessment capabilities is created.

### 3.4 Implementation

During this phase, iSpring software is utilised to transform PowerPoint presentations into HTML5 format, enabling the deployment of the teaching tool on the itch.io website, thereby ensuring accessibility through mobile browsers. The ultimate outcome can be obtained by accessing the following URL: https://cppspark.itch.io/functions/.

The hyperlink is thereafter distributed to a cohort of 33 students who express their willingness to test the application. The student cohort comprises individuals who have obtained scores below the mean in the recent C++ assessment. By utilising the hyperlink supplied, students will have the capability to conveniently access the tool using their personal laptops or mobile devices. Subsequently, respondents have the opportunity to respond to the pre and post-test questions as depicted in Figure 4.

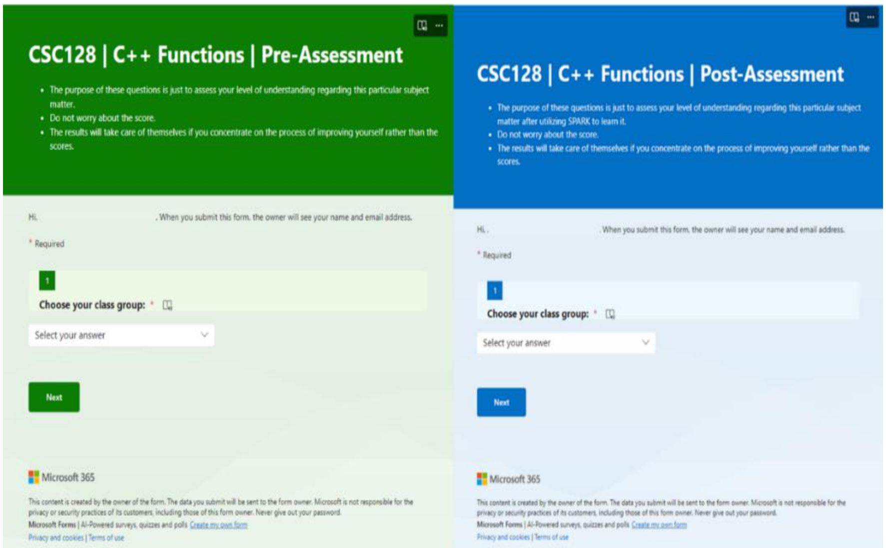


**Figure 4 Screenshot of Pre and Post-Test Created using Microsoft Form**

As depicted in Figure 4, it is recommended that students do the pre-test before engaging with the multimedia materials. Subsequently, they should proceed to the post-test, with both assessments having a maximum score of 28. Subsequently, the obtained outcomes are juxtaposed to scrutinise the academic achievements of the students about the given subject matter.

### 3.5 Evaluation

Based on pre and post-test results, Microsoft Excel is utilised to analyse the performance. The performance of the students on the C++ Function was compared and analysed using a normalised gain (N-gain) test, which involved scoring pre and post-test questions. The normalised gain, commonly expressed as ⟨g⟩, is frequently symbolised as:

$$\langle g \rangle = \frac{\text{PostAvg} - \text{PreAvg}}{28 - PreAvg}$$

The N-gain metric is employed as a quantitative measure to evaluate the comparative improvement achieved after an instructional intervention, such as the integration of a teaching technology. The total score of 28 signifies the highest attainable score for both the pre-test and the post-test. The calculation involves the division of the discrepancy between the mean post-test score and the mean pre-test score by the discrepancy between the highest attainable score and the mean pre-test score. The evaluative comparison conducted not only shed light on the efficiency of the instructional intervention but also yielded practical suggestions to improve the educational experience.

The following shows the steps to calculate the N-Gain score for each student involved in the treatment. The summary of the computations is presented in Table 3.

*Step 1: Calculate the Mean Pre-test Score*

The mean pre-test score was calculated by summing the pre-test scores of all participants and dividing by the total number of participants (N). For our dataset of 33 participants:

Mean Pre-test Score = (Sum of Pre-test Scores) / N

In the conducted investigation, the calculated average pre-test score was determined to be 14.3939.

*Step 2: Calculate the Mean Post-test Score*

Similarly, we calculated the mean post-test score using the post-test scores of 33 participants:

Mean Post-test Score = (Sum of Post-test Scores) / N

In the conducted investigation, the calculated average pre-test score was determined to be 22.5758.

*Step 3: Calculate the Gain for Each Participant*

The gain for each participant was calculated by subtracting their pre-test score from their post-test score:

Gain = Post-test Score – Pre-test Score

*Step 4: Calculate the Mean Gain*

The mean gain represents the average improvement across all participants:

Mean Gain = (Sum of Gains) / N

The computation results reveal a mean gain of 8.1818.

*Step 5: Calculate the Mean Maximum Possible Gain*

Mean Maximum Possible Gain = (Sum of Maximum Score – Pre-Test)/N

The computation results reveal a mean gain of 13.6161.

*Step 6: Calculate the N-Gain*

Finally, the N-Gain was calculated by normalizing the mean gain concerning the maximum possible gain:

N-Gain = (Mean Gain) / (Mean Maximum Possible Gain)

The computational findings indicate an average gain of 0.6336.

**Table 3: Summary of Results**

| Treatment | Value |
|---|---|
| Mean Pre-test Score | 14.3939 |
| Mean Post-test Score | 22.5758 |
| Mean Gain | 8.1818 |
| Mean Maximum Possible Gain | 13.6161 |
| N-Gain | 0.6336 |

## 4.0 RESULT AND DISCUSSION

In this section, the study presents the calculation and interpretation of the N-Gain as a metric for assessing the enhancement in scores after an educational intervention or treatment. The N-Gain metric facilitates the evaluation of the intervention's efficacy, taking into account the variability observed in the pre-test scores.

A cohort consisting of 33 students underwent a pre-assessment before engaging in the study of the topic C++ function. Upon acquiring knowledge about the subject matter, the students proceed to respond to the post-assessment. The scores obtained by students are calculated using the N-Gain Score methodology. The mean N-Gain score, computed for all students, was 0.6336. The analysis of the results indicated that a substantial proportion of students exhibited N-Gain scores that were within the range of moderate gains (0.3 to 0.7). The table displaying the classical N-Gain results is presented below.

**Table 4: The result of N-Gain criteria**

| Score Range | No of Students | Percentage | Criteria |
|---|---|---|---|
| g > 0.7 | 9 | 27.3 | High |
| g > 0.3, g <= 0.7 | 24 | 72.7 | Moderate |
| g <= 0.3 | 0 | 0 | Low |

According to the data presented in Table 4, it is evident that 27.3% of students met the high gain requirements, while 72.7% achieved moderate gain. Notably, no students fell into the poor gain category. The effectiveness of applying the teaching tool has been observed through an increase in the value of the pre-test and post-test. According to Hake (1999), a score more than 0.7 is considered a significant gain, while a score below 0.3 is considered a low gain. A moderately favourable effect on students' understanding of C++ functions was indicated by an average N-Gain score of 0.6336, falling within the medium gain range. The findings suggest that there was a significant increase in students' knowledge levels after utilising the tool, as evidenced by the average improvement observed from their initial baseline.

Although the medium gain exhibits promise, it also suggests the potential for greater improvement. Certain elements of the teaching tool may require further improvement. Potential enhancements to educational materials may encompass enhancing the simplicity of analogical explanations, integrating a wider array of multimedia components, or integrating multiple forms of assessments to accommodate a range of learning preferences. It is imperative to take into account variations among individuals. Not all students may have derived equal benefits from the teaching tool. Certain individuals may have attained significant advancements, although others may have encountered only minimal enhancements. Further examination of individual scores could yield valuable insights into particular sectors of the student population for which the tool demonstrates optimal performance.

The moderate level of achievement demonstrated by the teaching tool serves as an indication to educators regarding the possible advantages associated with the incorporation of multimedia and interactive tools into the curriculum. Although traditional teaching approaches continue to hold significance, incorporating innovative tools into instruction has the potential to optimise learning outcomes. It is also imperative to recognise the limitations of the research. Although the N-Gain score provides valuable insights, it mostly adopts a quantitative approach and may not fully encompass the complex aspects of the learning process. Additionally, it is important to consider external variables that may impact the outcomes, such as individuals' previous experience with similar tools or their unique study habits.

Considering the moderate level of increase achieved, it would be beneficial for future research endeavours to concentrate on incremental enhancements to the instrument, subsequently followed by a thorough reassessment. Furthermore, the inclusion of qualitative feedback from students has the potential to offer a more comprehensive understanding of the particular elements that they consider to be both advantageous and disadvantageous.

## 5.0 CONCLUSION

It is noteworthy that the N-Gain analysis considered the variability in pre-test scores among participants. This robust statistical approach ensures that the N-Gain value accurately reflects the improvement in scores, even in situations where participants begin with diverse levels of knowledge. The individual differences observed in N-Gain underscore the importance of personalized learning approaches and highlight the potential for tailoring instructional methods to cater to diverse learner needs. The educational significance of the observed improvement in scores cannot be understated. An enhanced understanding of C++ functions equips students with valuable skills and knowledge that are essential for success in computer programming. Furthermore, it lays a solid foundation for their academic journey and future professional development in the field of engineering.

The findings of our research hold practical implications for educators and instructional designers. The success of the teaching tool and e-content application in this study provides a blueprint for the development of similar resources in other educational contexts. Educators can leverage the power of multimedia and visualization to create engaging learning experiences that resonate with modern students. Despite the promising results, this study is not without limitations. The sample size and specific context of the study may limit the generalizability of the findings. Future research could explore the adaptability of similar interventions in diverse educational settings and examine the long-term retention of knowledge acquired through e-content applications.

In conclusion, our study demonstrates that a teaching tool and e-content application enriched with effective visualization techniques and multimedia elements can significantly enhance students' understanding of complex topics like C++ functions. The N-Gain analysis underscores the importance of innovative educational approaches in today's digital age and offers valuable insights for educators seeking to create engaging and impactful learning experiences. This research contributes to the evolving landscape of educational technology and pedagogy, paving the way for future advancements in the field of computer science education.

**ACKNOWLEDGEMENTS**

The authors wish to acknowledge the invaluable contributions of their colleagues and peers who provided encouragement and insightful discussions during the research process. We also extend our gratitude to the Universiti Teknologi MARA Cawangan Johor Kampus Pasir Gudang for providing access to essential research facilities.

## REFERENCES

Alabay, S., & Bastürk, M. (2021). Development, Implementation and Evaluation of E-Learning Materials for FFL with Adobe Captivate Software. *International Education Studies*, *14*(6), 59–70.

Amali, L. N., Kadir, N. T., & Latief, M. (2019). Development of e-learning content with H5P and iSpring features. *Journal of Physics: Conference Series*, *1387*(1), 12019.

Anak Yunus, C. C., & Hua, T. K. (2021). Exploring a gamified learning tool in the ESL classroom: The case of Quizizz. Journal of Education and e-Learning Research, 8(1), 103-108.

Anwar, M. S., Choirudin, C., Ningsih, E. F., Dewi, T., & Maseleno, A. (2019). Developing an interactive mathematics multimedia learning based on ispring presenter in increasing students' interest in learning mathematics. Al-Jabar: Jurnal Pendidikan Matematika, 10(1), 135-150.

Azad, A., & Smith, D. T. (2014). Teaching an introductory programming language in a general education course. Journal of Information Technology Education. Innovations in Practice, 13, 57.

Bekk, M., Eppmann, R., Klein, K., & Völckner, F. (2022). All that glitters is not gold: An investigation into the undesired effects of gamification and how to mitigate them through gamification design. *International Journal of Research in Marketing*, *39*(4), 1059–1081.

Berk, R. A. (2009). Multimedia teaching with video clips: TV, movies, YouTube, and mtvU in the college classroom. *International Journal of Technology in Teaching & Learning*, *5*(1).

Berssanette, J. H., & de Francisco, A. C. (2021). Cognitive load theory in the context of teaching and learning computer programming: A systematic literature review. *IEEE Transactions on Education*, *65*(3), 440–449.

Bottino, R., Chioccariello, A., & Freina, L. (2023). Developing Computational Fluency via Multimedia Stories. In Teaching Coding in K-12 Schools: Research and Application (pp. 63-79). Cham: Springer International Publishing.

Boukhechba, H., & Bouhania, B. (2019). Adaptation of instructional design to promote learning in traditional EFL classrooms: Adobe Captivate for e-learning content. *International Journal of*

*Education and Development Using Information and Communication Technology*, *15*(4), 151–164.
Dunican, E. (2002). *Making the analogy: Alternative delivery techniques for first year programming courses*.
Guzman, L. M., Pennell, M. W., Nikelski, E., & Srivastava, D. S. (2019). Successful integration of data science in undergraduate biostatistics courses using cognitive load theory. *CBE—Life Sciences Education*, *18*(4), ar49.
Hadza, C., Sesrita, A., & Suherman, I. (2020). Development of Learning Media Based on Articulate Storyline. *Indonesian Journal of Applied Research (IJAR)*, *1*(2), 80–85.
Hanika, S., & Guspatni, G. (2023). Development of learning media powerpoint-iSpring integrated with prompting questions on stoichiometry topics. Jurnal Pijar Mipa, 18(1), 57-64.
Haryana, M. R. A., Warsono, S., Achjari, D., & Nahartyo, E. (2022). Virtual reality learning media with innovative learning materials to enhance individual learning outcomes based on cognitive load theory. *The International Journal of Management Education*, *20*(3), 100657.
Ibrahim, N. S., Hasmy, H., Kapi, A. Y., & Suhaimi, A. (2018). The Usability of C++ Interactive Self-Assessment Quiz (i-SAQ). *International Journal of Human and Technology Interaction (IJHaTI)*, *2*(2), 67–74.
Jiménez Toledo, J. A., Collazos, C. A., & Ortega, M. (2021). Discovery model based on analogies for teaching computer programming. *Mathematics*, *9*(12), 1354.
Kapi, A. Y., Osman, N., Ramli, R. Z., Taib, J. M., & others. (2017). Multimedia education tools for effective teaching and learning. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *9*(2–8), 143–146.
Korn, O. (2022). Gamification in industrial production: An overview, best practices, and design recommendations. *Human-Technology Interaction: Shaping the Future of Industrial User Interfaces*, 251–270.
Linn, A., & Rea, P. (2016). *Undergraduate Medical Students as Active Participants and Co-Producers of E-Learning Tutorials*.
Narayan, R., Rodriguez, C., Araujo, J., Shaqlaih, A., & Moss, G. (2013). *Constructivism—Constructivist learning theory*.
Neureiter, D., Klieser, E., Neumayer, B., Winkelmann, P., Urbas, R., & Kiesslich, T. (2020). Feasibility of Kahoot! as a real-time assessment tool in (histo-) pathology classroom teaching. Advances in Medical Education and Practice, 695–705.
Paivio, A. (1991). Dual coding theory: Retrospect and current status. *Canadian Journal of Psychology/Revue Canadienne de Psychologie*, *45*(3), 255.
Pavithra, A., Aathilingam, M., & Prakash, S. M. (2018). Multimedia and its applications. *International Journal for Research & Development in Technology*, *10*(5), 271–276.
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.
Ramli, R. Z., Kapi, A. Y., & Osman, N. (2015). Visualization makes array easy. *Testing and Measurement: Techniques and Applications - Proceedings of the 2015 International Conference on Testing and Measurement: Techniques and Applications, TMTA 2015*.
Ramli, R. Z., Kapi, A. Y., & Osman, N. (2015, June). Visualization makes array easy. In Proceedings of the 2015 International Conference on Testing and Measurement: Techniques and Applications (TMTA'15). CRC Press (pp. 381-384).
Renkl, A., Hilbert, T., & Schworm, S. (2009). Example-based learning in heuristic domains: A cognitive load theory account. *Educational Psychology Review*, *21*(1), 67–78.
Romisa, F. (2023). PENGEMBANGAN MEDIA PEMBELAJARAN BERBASIS ANDROID MENGGUNAKAN ISPRING SUITE PADA MATA PELAJARAN INFORMATIKA MATERI PERANGKAT KERAS KOMPUTER. Jurnal Ilmiah Sains Teknologi Dan Informasi, 1(2), 17-23.
Salleh, S. M., Shukur, Z., & Judi, H. M. (2018). Scaffolding model for efficient programming learning based on cognitive load theory. *Int. J. Pure Appl. Math*, *118*(7), 77–83.
Shunkov, V., Shevtsova, O., Koval, V., Grygorenko, T., Yefymenko, L., Smolianko, Y., & Kuchai, O. (2022). Prospective Directions of Using Multimedia Technologies in the Training of Future Specialists. IJCSNS International Journal of Computer Science and Network Security, 22(6), 739.

https://doi.org/10.22937/IJCSNS.2022.22.6.93
Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education (TOCE)*, *13*(4), 1–64.
Swacha, J., & Szydłowska, J. (2023). Does Gamification Make a Difference in Programming Education? Evaluating FGPE-Supported Learning Outcomes. Education Sciences, 13(10), 984.
Theodoropoulos, A., & Lepouras, G. (2021). Augmented Reality and programming education: A systematic review. International Journal of Child-Computer Interaction, 30, 100335.
Van Merrienboer, J. J. G., & Sweller, J. (2005). Cognitive load theory and complex learning: Recent developments and future directions. *Educational Psychology Review*, *17*, 147–177.
Vikulova, L. G., Makarova, I. V., & Gerasimova, S. A. (2018). Features of iSpring suite learning platform for teaching foreign languages. Espacios, 39(20), 5-5.
Wang, C.-H., & Wu, K.-C. (2022). Interdisciplinary Learning of Low-Code Development Platform Programming with Dual Coding Theory-A Case Study of Agilepoint NX. *Journal of ICT, Design, Engineering and Technological Science*, 21–25.
Wulf, T. (2005). Constructivist approaches for teaching computer programming. *Proceedings of the 6th Conference on Information Technology Education*, 245–248.
Yunus, C. C. A., & Hua, T. K. (2021). Exploring a gamified learning tool in the ESL classroom: The case of Quizizz. *Journal of Education and E-Learning Research*, *8*(1), 103–108. https://doi.org/10.20448/JOURNAL.509.2021.81.103.108