International Journal of Human Resource and Workplace Studies (IJHRWS)

VOLUME 1 ISSUE 1 (2026)

PUBLISHED BY
E-PALLI PUBLISHERS, DELAWARE, USA

International Journal of Human Resource and Workplace Studies (IJHRWS)
Volume 1 Issue 1, Year 2026
https://journals.e-palli.com/home/index.php/ijhrws

Effect of Python-Integrated Mathematics Instruction on Mathematics Achievement and Mathematical Problem-Solving Skills among Senior Secondary School Students in Kwara State, Nigeria

Olurotimi David ADULOJU[1*], Lydia Oluwafunmilayo ADEDOTUN[2], Abiola Afolabi AKINGBEMISILU[1], Ademiotan Moriyike LALEYE[1], Margaret Foluso AYENI[2]

Article Information

Received: June 19, 2026
Accepted: August 26, 2026
Published: October 29, 2026

Keywords

Mathematical Problem-Solving Skills, Mathematics Achievement, Nigeria, Python-Integrated Mathematics Instruction, Quasi-Experimental Study, Secondary School Students

ABSTRACT

This study investigated the effect of Python-integrated Mathematics instruction on Mathematics achievement and mathematical problem-solving skills among Senior Secondary School II (SS II) students in Kwara State, Nigeria. A quasi-experimental pre-test–post-test non-equivalent control-group design was adopted. The sample comprised 80 SS II Mathematics students from two public secondary schools, with 40 students assigned to the experimental group and 40 to the control group. The experimental group received Python-integrated Mathematics instruction, while the control group was taught the same content using conventional instruction. Data were collected using the Mathematics Achievement Test (MAT) and Mathematical Problem-Solving Skills Test (PSST), with reliability coefficients of 0.82 and 0.80, respectively. Independent-samples t-tests examined baseline differences, while Analysis of Covariance (ANCOVA) assessed intervention effects while controlling for corresponding pre-test scores. Results showed no significant difference between the experimental and control groups in pre-test Mathematics achievement and mathematical problem-solving performance, indicating comparable baseline status. However, Python-integrated Mathematics instruction produced a statistically significant effect on Mathematics achievement, $F(1, 77) = 142.50$, $p < .001$, $\eta p^2 = .650$, and mathematical problem-solving skills, $F(1, 77) = 118.30$, $p < .001$, $\eta p^2 = .610$. The null hypotheses concerning the intervention effects were therefore rejected. The study concludes that Python-integrated Mathematics instruction can substantially enhance students' Mathematics achievement and problem-solving skills. It is recommended that Mathematics teachers be encouraged to integrate appropriately scaffolded Python-based activities into Mathematics instruction, while schools and curriculum planners should provide the necessary teacher training and technological resources to support effective implementation.

INTRODUCTION

Mathematics is a core subject in secondary education due to its importance in developing logical reasoning, analytical thinking, and problem-solving skills among learners. It also serves as a foundation for science, technology, engineering, and innovation. Despite its importance, students' achievement in Mathematics in Nigeria has remained persistently low, particularly in external examinations such as WAEC and NECO, where many students continue to perform below expectation (West African Examinations Council [WAEC], 2023). The persistent poor performance has been linked to the continued use of traditional teacher-centered instructional methods, which often limit students' active participation in the learning process. In such classrooms, students are mainly passive recipients of knowledge, making it difficult for them to understand abstract mathematical concepts and apply them to real-life situations. This gap between instruction and application contributes significantly to students' weak problem-solving abilities in Mathematics (Awofala, 2017).

In response to these challenges, educational researchers have emphasized the need for innovative instructional strategies that promote active learning and engagement. One such strategy is the integration of Information and Communication Technology (ICT) in teaching. ICT-based instruction has been shown to improve students' understanding of abstract concepts by enhancing visualization, interaction, and engagement in learning (UNESCO, 2021). As a result, ICT integration is increasingly becoming a global priority in education reform. More recently, attention has shifted toward programming-based instruction as a modern ICT approach in education. Programming helps learners develop computational thinking, logical reasoning, and structured problem-solving skills. Among programming languages, Python has gained global recognition due to its simplicity, readability, and wide application in education, data science, and artificial intelligence (Guo, 2007).

Empirical evidence suggests that programming activities can improve students' thinking skills and academic performance in STEM-related subjects. For instance, studies have shown that programming instruction enhances learners' ability to analyze problems, design solutions, and apply logical reasoning in different contexts (Grover & Pea, 2013). Similarly, integrating coding

[1] Department of Science Education, Faculty of Education, Adekunle Ajasin University, Akungba -Akoko, Ondo State, Nigeria
[2] Department of Science Education, Faculty of Education, Ekiti State University, Ado-Ekiti, Ekiti State, Nigeria
[*] Corresponding author's email: davidolurotimi@gmail.com

activities into classroom instruction has been reported to improve students' engagement and achievement in Mathematics-related tasks (Basu *et al.*, 2017). In Nigeria, studies have also reported that ICT-based instructional strategies significantly improve students' achievement in Mathematics compared to conventional methods (Adedeji *et al.*, 2018). However, most of these studies have focused on general ICT tools or instructional strategies such as computer-assisted instruction, with limited attention given to the use of Python programming as a teaching tool in secondary school Mathematics.

Furthermore, while students' academic achievement in Mathematics has been widely studied, less emphasis has been placed on higher-order learning outcomes such as problem-solving skills. Problem-solving is a critical component of Mathematics education because it enables learners to apply knowledge to unfamiliar situations, analyze problems systematically, and generate logical solutions (Polya, 1957). In modern education, problem-solving is considered essential for preparing students for real-world challenges and technological advancement. Given the growing importance of programming in education and the persistent challenges in Mathematics achievement among Nigerian secondary school students, there is a need to explore the effectiveness of integrating Python programming into Mathematics instruction. Therefore, this study investigates the effect of Python-integrated Mathematics instruction on Mathematics achievement and mathematical problem-solving skills among Senior Secondary School students in Kwara State, Nigeria.

Statement of the Problem

Despite the importance of Mathematics as a core subject in secondary education and its relevance to scientific and technological development, students' performance in the subject has continued to be a major concern in Nigeria. Evidence from external examination bodies such as the West African Examinations Council (WAEC) consistently shows that a significant proportion of secondary school students perform below the expected standard in Mathematics (WAEC, 2023). This persistent underachievement raises concerns among educators, researchers, and policymakers. Poor performance in Mathematics has been associated with several interacting factors, including instructional practices that may provide limited opportunities for active engagement, exploration and application of mathematical concepts, which often emphasize rote learning rather than conceptual understanding and application. In such teacher-centered classrooms, students are rarely engaged in active learning experiences that promote reasoning, exploration, and problem-solving. Consequently, many students struggle to apply mathematical concepts to real-life situations, resulting in weak problem-solving skills and low academic achievement (Awofala, 2017).

In response to these challenges, various innovative instructional approaches have been introduced, including ICT-based learning strategies. However, many of these approaches have focused on general computer-assisted instruction without adequately integrating programming tools that can require students to represent mathematical relationships, formulate procedures, test solutions and revise their reasoning. Although programming education has been widely recognized as a powerful tool for developing logical reasoning and problem-solving skills, its integration into Mathematics instruction in Nigerian secondary schools remains limited. In particular, Python programming, which is widely regarded as a simple and effective programming language for educational purposes, has not been sufficiently explored as an instructional tool for teaching Mathematics at the secondary school level in Nigeria. This is despite global evidence suggesting that programming-based instruction can enhance students' engagement, conceptual understanding, and problem-solving abilities in STEM-related subjects (Grover & Pea, 2013).

Furthermore, while previous studies have focused largely on students' academic achievement in Mathematics, less attention has been given to higher-order thinking skills such as problem-solving, which are essential for applying mathematical knowledge in unfamiliar and real-world contexts. This creates a gap in literature regarding the combined effect of programming-based instruction on both achievement and problem-solving skills among secondary school students. Therefore, the problem of this study is that despite the increasing relevance of programming in modern education and the persistent poor performance of students in Mathematics in Nigeria, there is limited empirical evidence on the effect of integrating Python programming into Mathematics instruction on students' achievement and problem-solving skills. It is against this background that this study seeks to investigate the effect of Python programming integration on students' achievement and problem-solving skills in Mathematics among Senior Secondary School students in Nigeria.

Purpose of the Study

The main purpose of this study is to examine the effect of Python-integrated Mathematics instruction on students' Mathematics achievement and mathematical problem-solving skills among senior secondary school students in Kwara State, Nigeria.

Specifically, the study seeks to:

1. determine the difference in the pre-test mean Mathematics achievement scores of students in the experimental and control groups before the intervention;
2. determine the difference in the pre-test mean mathematical problem-solving scores of students in the experimental and control groups before the intervention;
3. determine the effect of Python-integrated Mathematics instruction on students' post-test Mathematics achievement after controlling for their pre-test achievement scores;
4. determine the effect of Python-integrated

Mathematics instruction on students' post-test mathematical problem-solving scores after controlling for their pre-test problem-solving scores.

Research Questions

The following research questions will guide the study:

1. What difference exists in the pre-test mean Mathematics achievement scores of students in the experimental and control groups before the intervention?
2. What difference exists in the pre-test mean mathematical problem-solving scores of students in the experimental and control groups before the intervention?
3. What effect does Python-integrated Mathematics instruction have on students' Mathematics achievement after controlling for their pre-test achievement scores?
4. What effect does Python-integrated Mathematics instruction have on students' mathematical problem-solving skills after controlling for their pre-test problem-solving scores?

Research Hypotheses

The following null hypotheses will be tested at the 0.05 level of significance:

$\mathbf{H_{01}}$: There is no significant difference in the pre-test mean Mathematics achievement scores of students in the experimental and control groups before the intervention.

$\mathbf{H_{02}}$: There is no significant difference in the pre-test mean mathematical problem-solving scores of students in the experimental and control groups before the intervention.

$\mathbf{H_{03}}$: There is no significant effect of Python-integrated Mathematics instruction on students' Mathematics achievement after controlling for their pre-test achievement scores.

$\mathbf{H_{04}}$: There is no significant effect of Python-integrated Mathematics instruction on students' mathematical problem-solving skills after controlling for their pre-test problem-solving scores.

Significance of the Study

The findings of this study are expected to be significant to students, teachers, curriculum planners, educational researchers, and policymakers in Nigeria.

First, the study will be beneficial to secondary school students by exposing them to innovative instructional approaches such as Python programming integration in Mathematics. This approach is expected to enhance their understanding of mathematical concepts, improve their academic achievement, and strengthen their problem-solving skills. By engaging students in interactive and practical learning experiences, the study will help bridge the gap between abstract mathematical concepts and real-life applications.

Second, the study will be of importance to Mathematics teachers by providing them with evidence-based instructional strategies that can improve classroom practice. The findings will encourage teachers to adopt programming-based instructional methods that promote active learning, critical thinking, and learner engagement. This may also enhance teachers' professional competence in integrating ICT tools into Mathematics instruction.

Third, the study will be useful to curriculum planners and educational developers such as the Nigerian Educational Research and Development Council (NERDC). The results will provide empirical evidence on the effectiveness of integrating programming (Python) into Mathematics instruction, which may inform future curriculum reforms aimed at strengthening the integration of programming and technology-supported mathematical learning in secondary education.

Fourth, the study will contribute to educational researchers and scholars by adding to existing literature on ICT integration, programming education, and Mathematics pedagogy. It will also serve as a reference point for future studies on the use of Python programming and other computational tools in enhancing students' learning outcomes in STEM education.

Finally, the study will be significant to education policymakers and government agencies by providing evidence that can guide policy decisions on the integration of programming education into secondary school curricula. This may support national efforts toward improving technology-supported mathematical learning, problem-solving and innovation-oriented education in Nigeria.

Overall, the study is expected to contribute to improving the quality of Mathematics education by demonstrating the effectiveness of Python programming integration in enhancing students' achievement and problem-solving skills.

Delimitation of the Study

This study is delimited to SS2 students offering Mathematics in selected public secondary schools in Kwara State, Nigeria. It focuses on the effect of Python-integrated Mathematics instruction on students' Mathematics achievement and mathematical problem-solving skills. The study is restricted to selected SS2 Mathematics topics considered suitable for integration with Python programming. The study employs a pre-test–post-test non-equivalent control-group quasi-experimental design involving intact classes assigned to experimental and control conditions. The findings are therefore interpreted within the context of the selected schools and participants in Kwara State.

LITERATURE REVIEW

Introduction

This section reviews literature relevant to the effect of Python-integrated Mathematics instruction on students' Mathematics achievement and mathematical problem-solving skills. The review is organized into four major components: conceptual review, theoretical review, empirical review and summary of the literature and identified gap.

The conceptual review examines the concepts of

Mathematics instruction, Python-integrated Mathematics instruction, Mathematics achievement and mathematical problem-solving skills. The theoretical review considers Constructivist Learning Theory as the principal theoretical foundation, Cognitive Load Theory as a supporting perspective, and Polya's problem-solving framework as the outcome-specific framework for mathematical problem-solving. The empirical review critically examines relevant studies on technology-supported Mathematics instruction, activity-oriented Mathematics instruction and programming-integrated Mathematics learning, with particular attention to evidence from Nigeria and Kwara State. The chapter concludes by identifying the gap that the present study seeks to address.

Conceptual Review

Concept of Mathematics Instruction

Mathematics instruction refers to the planned processes through which mathematical knowledge, concepts, procedures, representations and problem-solving strategies are presented, explored and applied in the classroom. Effective Mathematics instruction extends beyond the transmission of formulas and procedures. It should provide learners with opportunities to understand mathematical relationships, represent ideas in different forms, reason about mathematical situations and apply knowledge to problems. This broader conception is important because mathematical proficiency involves more than procedural accuracy. Recent theoretical work on mathematical proficiency emphasizes that effective instruction should support students' understanding of mathematical concepts and their ability to reason with and apply mathematical knowledge rather than merely reproduce procedures (Ngu & Phan, 2024).

In conventional Mathematics classrooms, instruction may involve teacher explanation, worked examples, guided practice and independent exercises. These approaches can be useful, particularly when students require explicit explanation of unfamiliar procedures. However, when instruction relies predominantly on explanation and repetitive exercises, opportunities for students to explore mathematical relationships, test conjectures and examine alternative solution strategies may be restricted.

Consequently, contemporary Mathematics education increasingly recognizes the importance of instructional approaches that actively engage students in mathematical activity. A 2024 meta-analysis of constructivist approaches to Mathematics education, for example, found evidence across studies of constructivist approaches being used to improve students' mathematical abilities, although the magnitude of effects varied across studies. The implication is not that teacher-directed instruction should be completely abandoned. Rather, effective Mathematics instruction may require an appropriate combination of explanation, modelling, guided practice, exploration and application. This becomes particularly relevant when technology is introduced into Mathematics instruction.

Technology-Supported Mathematics Instruction

Technology-supported Mathematics instruction involves the deliberate use of digital technologies to facilitate the teaching and learning of mathematical concepts, procedures, representations and applications. Such technologies may include mathematical software, computer applications, graphing tools, interactive simulations and programming environments. Technology can provide mathematical representations and computational facilities that are difficult or time-consuming to produce manually. It can also allow students to experiment with mathematical relationships, obtain rapid feedback and examine the consequences of changing variables or procedures. However, technology should not be regarded as inherently instructional. Its educational value depends on how it is incorporated into learning activities. This distinction is particularly important for the present study because the intervention is not simply the provision of computers or digital resources. Rather, Python is integrated into mathematical tasks so that students use programming procedures as part of learning mathematical concepts and solving mathematical problems.

Evidence from Kwara State provides a useful local illustration. Salami and Spangenberg (2024) investigated ICT facilities and Mathematics performance among secondary school students in Kwara State using a quasi-experimental pre-test/post-test non-equivalent control-group design. Their sample consisted of 80 students from two secondary schools. Students exposed to ICT-supported Mathematics instruction demonstrated improved achievement compared with those taught through the traditional approach.

The study is particularly relevant because it demonstrates that technology-supported Mathematics instruction can be investigated experimentally within the Kwara State context. Nevertheless, the intervention involved ICT facilities generally and did not examine programming as the instructional medium. It also focused primarily on Mathematics performance rather than mathematical problem-solving as a separate outcome. Consequently, its findings provide contextual support for the present study but do not directly establish the effectiveness of Python-integrated Mathematics instruction.

Python-Integrated Mathematics Instruction

Python-integrated Mathematics instruction refers to the deliberate incorporation of the Python programming language into mathematical teaching and learning activities. In this approach, Python is not taught merely as a computer-programming subject. Instead, students use Python to represent, investigate, calculate, test or solve mathematical problems while learning the underlying mathematical concepts. The distinction between using technology in Mathematics and integrating programming into Mathematics is important. In conventional technology-supported instruction, students may use software to view a graph, obtain a calculation or

manipulate an existing representation. In programming-integrated instruction, students construct or modify procedures that instruct the computer to perform mathematical operations.

This requires students to make mathematical relationships explicit. For example, students may need to identify variables, translate mathematical relationships into computational procedures, sequence operations, execute the procedure, inspect the output and revise the procedure when errors occur. Recent research provides empirical support for examining this instructional possibility. Rais and Xuezhi (2024) investigated Python programming integrated into Mathematics learning among Grade X students in Indonesia. Their intervention used Python through the Pydroid 3 environment on smartphones and involved Mathematics topics concerning systems of linear equations. The instructional sequence required students to understand the mathematical concept, solve problems manually, represent mathematical procedures through Python, execute the programs, inspect errors and compare computational outputs with their manual solutions.

The study is particularly relevant because it demonstrates how Python can be incorporated into Mathematics rather than treated as an independent programming activity. The researchers reported increased student motivation and positive learning experiences, although their methodology differed substantially from the present study. The intervention involved 92 Grade X students, used non-parametric analyses and examined motivation and learning outcomes rather than employing the ANCOVA-based quasi-experimental approach adopted in the present study. Therefore, the study provides evidence of the feasibility and educational potential of Python-integrated Mathematics instruction but does not eliminate the need for further quasi-experimental research.

Munthe and Naalsund (2024) further emphasized that programming can be incorporated into mathematical problem tasks in ways that maintain Mathematics as the primary learning objective. Their work specifically addressed the design of mathematical programming problems and demonstrated that programming tasks can be designed so that students' attention remains focused on mathematical learning rather than programming syntax alone. This observation has direct implications for the present study. The effectiveness of Python-integrated Mathematics instruction depends not simply on exposing students to Python but on designing mathematical activities in which Python serves the mathematical learning objective.

Mathematics Achievement

Mathematics achievement refers to the level of knowledge and competence demonstrated by students in relation to specified mathematical content and instructional objectives. It is commonly measured through achievement tests designed around the concepts, procedures and applications taught during instruction. In the present study, Mathematics achievement represents students' performance on the Mathematics Achievement Test administered before and after the intervention. The pre-test provides information concerning students' baseline achievement, while the post-test provides evidence of their achievement following exposure to the instructional treatment.

Mathematics achievement is an important outcome because it provides evidence of students' mastery and application of curriculum-related mathematical knowledge. However, achievement scores alone may not adequately reveal students' ability to apply mathematical knowledge strategically to unfamiliar situations. This is one reason the present study examines mathematical problem-solving skills as a second outcome. The distinction between the two outcomes is important. A student may acquire mathematical procedures sufficiently to perform well on structured achievement-test items but still experience difficulty when a problem requires interpretation, representation, selection of a strategy or evaluation of an unfamiliar solution.

Mathematical Problem-Solving Skills

Mathematical problem-solving skills refer to students' ability to understand mathematical problems, identify relevant information, represent the problem appropriately, develop and implement solution strategies, and evaluate the reasonableness of the resulting solution. Problem-solving is therefore more than performing calculations. It involves the coordinated use of mathematical knowledge, reasoning, representation, strategy selection and evaluation. Recent literature continues to identify problem-solving as an important component of Mathematics learning. Wahab *et al.* (2024), in a systematic review of 48 studies conducted between 2019 and 2023, examined students' difficulties across the four stages of Polya's framework. The review found difficulties in understanding problems, devising plans, carrying out procedures and looking back to evaluate solutions.

The finding is particularly relevant to the present study because it demonstrates that students' difficulties may occur at different stages of the problem-solving process. Consequently, an intervention that provides repeated opportunities for students to represent, execute, inspect and revise mathematical procedures may potentially support problem-solving development. However, such an effect should be established empirically rather than assumed. Python-integrated Mathematics instruction may provide an environment for students to engage in some of these processes, but the present study specifically tests whether this translates into measurable improvement in mathematical problem-solving performance.

Relationship between Mathematics Achievement and Mathematical Problem-Solving

Mathematics achievement and mathematical problem-solving are closely related but should not be treated as identical constructs. Mathematics achievement primarily reflects students' acquisition and application

of mathematical knowledge specified by the instructional content. Mathematical problem-solving, in contrast, places greater emphasis on the application of mathematical knowledge to situations in which the solution path may not be immediately apparent. The two outcomes can therefore complement one another. Improvement in Mathematics achievement may indicate improved mastery of mathematical content, while improvement in problem-solving may indicate enhanced capacity to apply that knowledge strategically. This distinction provides a strong justification for examining both outcomes in the present study. If Python-integrated Mathematics instruction improves achievement but not problem-solving, its contribution may be primarily related to content mastery. If it improves both outcomes, the evidence would suggest broader educational value.

Theoretical Review

Constructivist Learning Theory

Constructivist Learning Theory provides the principal theoretical foundation for the present study. Constructivism views learning as an active process in which learners construct and reorganize knowledge through interaction with tasks, experiences and prior knowledge rather than simply receiving information from a teacher. The relevance of constructivism to Mathematics education lies in its emphasis on students' active engagement with mathematical ideas. Mathematics learning can involve exploring relationships, testing conjectures, comparing representations, interpreting results and revising previous understandings. Recent evidence continues to support investigation of constructivist approaches in Mathematics education. Jumaida *et al.* (2024), in a meta-analysis of constructivist approaches involving 20 publications, examined their effects on students' mathematical abilities. Although the reported effects varied, the study provides contemporary evidence that constructivist approaches remain relevant to Mathematics instruction. Python-integrated Mathematics instruction is compatible with this perspective because students can actively construct mathematical procedures rather than simply observe completed solutions. A typical learning cycle can be represented as:

Mathematical concept → mathematical representation → Python procedure → execution → output → interpretation → revision

For example, students may first formulate a mathematical relationship manually, translate it into a Python procedure, execute the procedure and compare the computational output with their mathematical expectation. When discrepancies occur, students can investigate whether the problem lies in their mathematical reasoning, representation or programming procedure. Such activities provide opportunities for students to interact directly with mathematical ideas and receive feedback from their actions. Nevertheless, constructivism does not imply that students should be left to discover mathematical knowledge without guidance. Programming may introduce unfamiliar syntax and procedures, particularly for students with little previous experience. Therefore, teacher scaffolding remains essential. This point is supported by Rais and Xuezhi (2024), whose Python-integrated Mathematics intervention involved explicit teacher facilitation, guidance with programming difficulties and comparison between manual mathematical solutions and Python outputs.

Implication for the present study

Constructivism provides a theoretical explanation for why Python-integrated Mathematics instruction may influence Mathematics achievement and mathematical problem-solving. By engaging students in representing, testing, interpreting and revising mathematical procedures, the intervention may provide richer opportunities for active construction and application of mathematical knowledge.

Cognitive Load Theory

Cognitive Load Theory (CLT), associated particularly with Sweller, Ayres and Kalyuga, provides a supporting explanation concerning the cognitive demands involved in Python-integrated Mathematics instruction. The theory is based on the limited capacity of working memory and emphasizes the importance of instructional designs that manage unnecessary cognitive demands while supporting the acquisition of knowledge structures. This perspective is especially relevant to programming-integrated Mathematics because students may simultaneously need to understand mathematical concepts and learn how to express those concepts through programming syntax. Consequently, Python may produce both opportunities and challenges. On the one hand, computational tools can reduce repetitive calculations and provide immediate feedback. On the other hand, programming syntax may impose additional cognitive demands on novice learners. This concern is reflected in Rais and Xuezhi's (2024) study. The researchers reported that some students initially experienced difficulty understanding and executing programming procedures and required teacher guidance, manuals and peer support. The implication is that Python-integrated instruction should be carefully sequenced. Appropriate strategies may include:

- introducing simple mathematical programming procedures before complex tasks;
- providing worked examples and model code;
- gradually increasing task complexity;
- connecting programming commands directly to mathematical concepts;
- providing immediate feedback;
- requiring students to compare computational outputs with mathematical solutions; and
- providing teacher scaffolding when students encounter programming difficulties.

Recent research also continues to demonstrate the relevance of instructional design and scaffolding within Cognitive Load Theory. Van Nooijen *et al.* (2024), for example, examined expert scaffolding of problem-

solving tasks from a Cognitive Load Theory perspective.

Implication for the present study

Cognitive Load Theory does not imply that Python inherently improves learning or reduces cognitive load. Instead, it suggests that the design and scaffolding of Python-integrated activities are important determinants of whether programming supports or interferes with mathematical learning.

Polya's Mathematical Problem-Solving Framework

Polya's framework provides the outcome-specific framework for the mathematical problem-solving component of the present study. Polya conceptualized mathematical problem-solving as a process involving four major stages:

1. Understanding the problem
2. Devising a plan
3. Carrying out the plan
4. Looking back

The first stage involves identifying what is known, what is unknown and what relationships exist in the problem. The second involves determining an appropriate strategy. The third requires implementing the selected strategy, while the fourth involves checking and interpreting the solution.

Recent systematic research continues to demonstrate the relevance of this framework. Wahab *et al.* (2024) reviewed 48 studies and found that students experience difficulties across all four stages. Problems included conceptual misunderstandings and weak interpretation during problem understanding, difficulties with mathematical modelling and planning, procedural errors during execution, and inadequate evaluation during the looking-back stage. Polya's framework corresponds meaningfully with Python-integrated Mathematics activities.

Interpretation of Table 1

Table 1 illustrates a conceptual correspondence between Polya's problem-solving stages and activities that may occur during Python-integrated Mathematics instruction. The correspondence does not imply that programming and mathematical problem-solving are identical processes. Rather, Python can provide an environment in which planning, sequencing, implementation, verification and revision become explicit components of mathematical activity. This is particularly relevant because programming-integrated Mathematics tasks can require students to identify a mathematical problem, formulate a procedure, implement it, examine the output and revise errors. Munthe and Naalsund (2024) similarly emphasized the importance of designing programming tasks in Mathematics in ways that preserve mathematical exploration and discussion as central learning activities.

Table 1: Correspondence between Polya's Problem-Solving Stages and Python-Integrated Mathematics Activities

Polya's stage	Python-integrated Mathematics activity
Understanding the problem	Identify known quantities, unknowns and mathematical relationships
Devising a plan	Formulate mathematical steps or an algorithm
Carrying out the plan	Translate and execute the mathematical procedure in Python
Looking back	Inspect the output, verify the solution, identify errors and revise the procedure

Implication for the present study

Polya's framework provides the conceptual basis for interpreting mathematical problem-solving as a multidimensional process rather than as simple computational accuracy. It therefore supports the use of a separate mathematical problem-solving test alongside the Mathematics Achievement Test.

Integrated Theoretical Framework

The three perspectives adopted in this study perform different but complementary functions.

Interpretation of Table 2

Table 2 shows that the three perspectives should not be viewed as competing theories. Constructivism provides the principal explanation for how students may actively construct mathematical knowledge through Python-supported activities. Cognitive Load Theory explains why the programming tasks require appropriate sequencing and scaffolding. Polya's framework provides a specific structure for understanding the mathematical problem-solving outcome. Together, the perspectives provide a coherent theoretical basis for examining whether Python-

Table 2: Integrated Theoretical Explanation of the Expected Effects of Python-Integrated Mathematics Instruction

Perspective	Main mechanism	Relevance to present study
Constructivist Learning Theory	Active construction, exploration, feedback and revision	Explains how Python-supported activities may facilitate Mathematics learning
Cognitive Load Theory	Management of task complexity and instructional scaffolding	Explains the importance of carefully designed Python activities
Polya's Problem-Solving Framework	Understanding, planning, execution and evaluation	Provides the framework for examining mathematical problem-solving

integrated Mathematics instruction can improve students' Mathematics achievement and mathematical problem-solving skills.

Conceptual Framework of the Study

The conceptual framework presents the expected relationship between Python-integrated Mathematics instruction and the two learning outcomes examined in the study. The independent variable is Python-integrated Mathematics instruction, while the dependent variables are Mathematics achievement and mathematical problem-solving skills. Python-integrated Mathematics instruction represents the deliberate use of Python programming as part of the teaching and learning of selected Mathematics topics. Rather than being taught as an independent programming activity, Python is incorporated into mathematical tasks through which students represent mathematical relationships, perform computations, explore patterns, test procedures, interpret results and revise their solutions.

The framework proposes that exposure to Python-integrated Mathematics instruction may influence students' Mathematics achievement and mathematical problem-solving skills. Mathematics achievement represents students' mastery and application of the mathematical content covered during the intervention, while mathematical problem-solving skills represent their ability to understand mathematical problems, formulate appropriate strategies, carry out solution procedures and evaluate their solutions. The framework is therefore presented as follows:

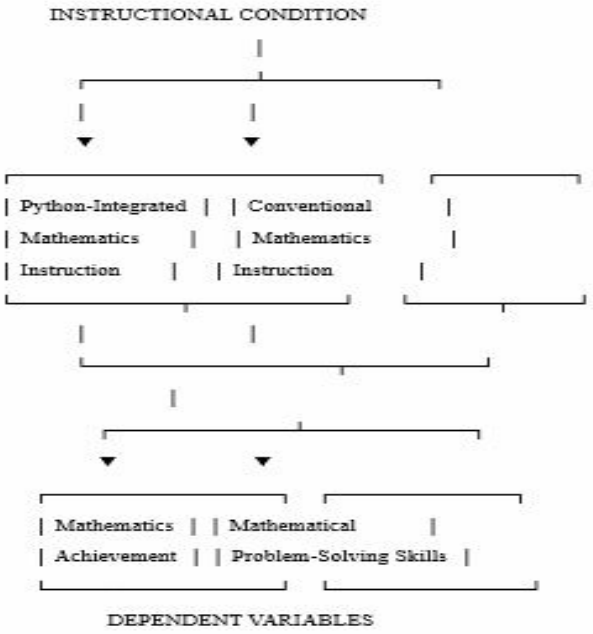

Figure 1: Conceptual Framework for the Effect of Python-Integrated Mathematics Instruction on Mathematics Achievement and Mathematical Problem-Solving Skills

Interpretation of Figure 1

Figure 1 illustrates the conceptual relationship underlying the study. Python-integrated Mathematics instruction constitutes the independent variable, while Mathematics achievement and mathematical problem-solving skills constitute the dependent variables. The framework proposes that integrating Python into Mathematics instruction provides students with opportunities to represent mathematical relationships computationally, perform and test mathematical procedures, examine results and revise their solutions. These learning activities are expected to contribute to students' mastery of mathematical content and their ability to apply mathematical knowledge in solving mathematical problems. However, the arrows in the framework represent the expected relationship to be investigated, rather than a predetermined finding. Whether Python-integrated Mathematics instruction produces statistically significant differences in Mathematics achievement and mathematical problem-solving skills is determined empirically through the quasi-experimental analysis.

Empirical Review

ICT-Supported Mathematics Instruction in Kwara State

Salami and Spangenberg (2024) investigated the impact of ICT facilities on Mathematics performance among secondary school students in Kwara State. The researchers employed a quasi-experimental pre-test/post-test non-equivalent control-group design involving 80 students selected from two secondary schools. The experimental group received Mathematics instruction using ICT facilities, while the control group received traditional instruction. ANCOVA was used to test the hypotheses.

The researchers reported improved Mathematics achievement among students exposed to ICT-supported instruction. The study is particularly relevant because it provides recent local evidence that technology-supported Mathematics instruction can produce measurable differences in achievement. Its quasi-experimental design also bears methodological similarity to the present study. However, the intervention was based on ICT facilities generally rather than Python programming. Moreover, the study focused on Mathematics performance and did not examine mathematical problem-solving skills as a separate outcome. The study therefore supports the broader rationale for technology-supported Mathematics instruction but leaves unanswered whether programming-integrated instruction produces similar effects.

Activity-Based Mathematics Instruction in Kwara State

Omolafe, *et al.*, (2024) examined the effect of activity-based teaching on secondary students' academic performance in Mathematics in Ilorin, Kwara State. The study employed a quasi-experimental non-equivalent control-group design and compared students taught Binary Numbers using an activity-based approach with those taught using a conventional method. The authors reported differences in students' academic performance following the intervention. The study demonstrates the relevance of active instructional approaches within the Kwara State context. However, the intervention was activity-based rather than programming-based, and the outcome focused on academic performance rather than mathematical problem-solving skills. It should also be noted that the 2023 publication by Omolafe and Okikiayo reports the earlier version of this activity-based study. It should therefore not be counted as a separate independent empirical study from the 2024 publication.

Python-Integrated Mathematics Instruction

Rais and Xuezhi (2024) investigated Python programming integration into Mathematics learning among Grade X students in Indonesia. The study involved 92 students from three schools implementing the Merdeka Belajar curriculum. Students used Pydroid 3 on smartphones to learn mathematical content involving systems of linear equations. The instructional sequence included manual mathematical work, Python programming, error identification, teacher facilitation and comparison between manual solutions and program outputs. The study reported increased student motivation and positive engagement with Python-supported Mathematics learning. Students also demonstrated increased familiarity with mathematical concepts such as variables, coefficients and equations.

The study is highly relevant to the present research because it demonstrates that Python can be integrated directly into Mathematics instruction rather than being taught independently as programming. However, several limitations reduce its direct comparability with the present study. The study was conducted outside Nigeria, used a different research design and statistical approach, focused on motivation and learning outcomes, and did not examine mathematical problem-solving as a distinct dependent variable using the ANCOVA structure adopted in the present research. Thus, while Rais and Xuezhi provide important international evidence concerning the feasibility and potential of Python-integrated Mathematics instruction, their findings cannot be directly generalized to Senior Secondary School students in Kwara State.

Programming and Mathematical Problem-Solving

Programming has also been investigated in relation to mathematical problem-solving. Munthe and Naalsund (2024) examined the design of mathematical programming problems and emphasized that programming tasks can be deliberately designed so that mathematical learning remains central. Their work suggests that programming may provide opportunities for mathematical exploration, discussion and examination of the limitations of computational procedures. Similarly, Rais and Xuezhi (2024) incorporated problem-solving activities into Python-supported Mathematics learning. Their instructional process required students to formulate mathematical models, solve them manually, translate them into Python procedures, execute the code and examine errors and outputs.

These studies provide conceptual and empirical support for the possibility that programming can be integrated into problem-oriented Mathematics learning. Nevertheless, neither establishes the specific effect of Python-integrated instruction on a separately measured mathematical problem-solving outcome among Nigerian Senior Secondary School students.

Critical Synthesis of Empirical Literature

Interpretation of Table 3

The reviewed studies indicate that technology-supported, activity-based and programming-integrated approaches have potential to enrich Mathematics learning. However, the evidence remains fragmented. The strongest local evidence comes from Kwara State, but the local studies examined general ICT or activity-based instruction rather than Python-integrated Mathematics. The international Python study demonstrates the feasibility of integrating Python with mathematical learning but differs from the present study in context, design and outcome measures. Research on mathematical problem-solving further demonstrates the importance of examining students' ability to understand, plan, execute and evaluate mathematical solutions. The literature therefore provides a rationale for the present study without providing a definitive answer to its research question.

Table 3: Selected Empirical Studies Relevant to the Present Study

Study	Context	Intervention/ focus	Design/ evidence	Main finding	Limitation relative to present study
Salami & Spangenberg (2024)	Kwara State, Nigeria	ICT-supported Mathematics	Quasi-experimental; N = 80	ICT-supported instruction improved Mathematics achievement	General ICT, not Python; no separate problem-solving outcome
Omolafe *et al.* (2024)	Ilorin, Kwara State	Activity-based Mathematics	Quasi-experimental	Activity-based instruction influenced Mathematics performance	Not programming-based; problem-solving not separately measured
Rais & Xuezhi (2024)	Indonesia	Python-integrated Mathematics	Quantitative intervention; N = 92	Python-supported Mathematics produced positive learning/ motivation outcomes	Different national context and design; problem-solving not examined as a distinct outcome
Munthe & Naalsund (2024)	International	Mathematical programming problems	Design/ research-based study	Programming tasks can support mathematical exploration	Focused on task design rather than a direct comparative intervention
Wahab *et al.* (2024)	International literature	Mathematical problem-solving	Systematic review; 48 studies	Students experience difficulties across Polya's stages	Does not examine Python-integrated instruction

Research Gap

The reviewed literature reveals three important gaps.

First, an instructional gap exists. Existing Nigerian evidence has examined technology-supported Mathematics instruction, but there is limited empirical evidence specifically concerning Python-integrated Mathematics instruction at the Senior Secondary School level.

Second, a contextual gap exists. Recent evidence from Kwara State demonstrates the potential of ICT-supported and activity-based Mathematics instruction, but these studies have not adequately examined Python as the instructional medium. Salami and Spangenberg (2024) investigated ICT facilities, while Omolafe *et al.* (2024) examined activity-based instruction.

Third, an outcome gap exists. Existing studies have frequently focused on Mathematics achievement or general academic performance. Relatively less attention has been given to examining Mathematics achievement and mathematical problem-solving skills simultaneously within a Python-integrated intervention.

Although Rais and Xuezhi (2024) provide relevant international evidence concerning Python-integrated Mathematics learning, their study was conducted in Indonesia and employed different methodological and outcome structures. Consequently, its findings cannot be assumed to apply directly to Nigerian Senior Secondary School students.

The present study addresses these gaps by investigating the effect of Python-integrated Mathematics instruction on both Mathematics achievement and mathematical problem-solving skills among Senior Secondary School students in Kwara State, Nigeria. The study further strengthens the evidence by using a pre-test–post-test non-equivalent control-group quasi-experimental design and statistically controlling for students' baseline performance through ANCOVA.

Summary of the Literature Review

The literature reviewed in this chapter establishes that Mathematics learning requires more than the acquisition of computational procedures. Students need opportunities to understand mathematical concepts, apply mathematical knowledge and solve unfamiliar problems. Technology-supported instruction can provide additional opportunities for mathematical exploration, although its effectiveness depends on the manner in which technology is integrated into classroom activities.

Python-integrated Mathematics instruction represents a more specific form of technology-supported instruction in which students use programming procedures to represent, execute, test and examine mathematical relationships. Recent international evidence demonstrates that Python can be incorporated into Mathematics learning, while recent Nigerian evidence demonstrates the potential of technology-supported and activity-based Mathematics instruction in Kwara State. However, these strands of evidence have not adequately established the effect of Python-integrated Mathematics instruction on both Mathematics achievement and mathematical problem-solving skills among Nigerian Senior Secondary School students.

The theoretical review indicates that Constructivist Learning Theory provides the principal explanation

for the active construction of mathematical knowledge through Python-supported activities. Cognitive Load Theory provides a complementary explanation concerning the need to manage the additional cognitive demands associated with programming. Polya's problem-solving framework provides the specific structure for understanding mathematical problem-solving as a process of understanding, planning, execution and evaluation.

Taken together, the conceptual, theoretical and empirical literature establishes a clear basis for the present investigation. The study therefore addresses an identifiable instructional, contextual and outcome-related gap by examining whether Python-integrated Mathematics instruction produces superior Mathematics achievement and mathematical problem-solving performance compared with conventional Mathematics instruction among Senior Secondary School students in Kwara State, Nigeria.

MATERIALS AND METHODS

Research Design

This study adopts a quasi-experimental research design, specifically the pre-test, post-test non-equivalent control group design. This design is appropriate because intact classes were used, and random assignment of individual students is not possible in school settings.

The design involves two groups: Experimental group (taught Mathematics using Python programming integration) and Control group (taught Mathematics using conventional teaching method), Both groups were pre-tested before treatment and post-tested after the instructional period to determine the effect of the intervention.

Area of the Study

The study will be conducted in Kwara State, Nigeria. Kwara State is chosen because it has a mix of urban and rural secondary schools with varying levels of ICT exposure, making it suitable for examining the effect of integrating Python programming into Mathematics instruction.

Population of the Study

The population of the study comprises all Senior Secondary School II (SS2) students offering Mathematics in public secondary schools in Kwara State.

Sample and Sampling Technique

The sample for this study consisted eighty (80) Senior Secondary Two (SS2) mathematics students in Kwara State, Nigeria.

A multi-stage sampling procedure was adopted to select the sample size for the study:

Stage One: Out of the three senatorial districts in Kwara State (Kwara Central, Kwara North, and Kwara South), Kwara Central Senatorial District was randomly selected using a simple random sampling technique (balloting).

Stage Two: From the local government areas (LGAs) within Kwara Central, Ilorin West Local Government Area was randomly selected.

Stage Three: Two senior secondary schools within the selected LGA were purposively selected. The purposive selection was based on the availability of adequate computer ICT laboratory facilities and infrastructure required for the Python-integrated programming instruction.

Stage Four: The two selected schools were randomly assigned to the experimental and control groups. Finally, a sample of students from each of the two intact SS2 classes, each comprising approximately 40 students, were selected from each school, yielding a total sample size of 80 students (40 in the experimental group and 40 in the control group) who participated in the study.

Research Instrumentation

Two instruments were used for data collection: the Mathematics Achievement Test (MAT) and the Mathematical Problem-Solving Skills Test (PSST). The MAT was developed to measure students' achievement in the Mathematics topics covered during the intervention, while the PSST was developed to assess students' ability to understand, represent, plan and solve non-routine mathematical problems related to the instructional content.

Although both instruments were objectively scored, they measured different constructs. The MAT focused primarily on students' acquisition and application of the Mathematics content taught during the intervention, whereas the PSST focused on students' ability to interpret non-routine mathematical situations, select appropriate solution strategies, execute those strategies and evaluate the resulting solutions. Thus, the PSST was not intended to duplicate the content coverage of the MAT but to assess the process-oriented application of mathematical knowledge to unfamiliar problems.

Validity and Reliability of the Instruments

To ensure the quality, accuracy, and dependability of the data collected, both instruments underwent rigorous validation and reliability testing.

Validity of the Instruments

Face and Content Validity: The draft copies of the Mathematics Achievement Test (MAT) and the Problem-Solving Skills Test (PSST), along with the lesson plans and instructional materials, were subjected to expert scrutiny. They were evaluated by three senior secondary mathematics educators and two measurement and evaluation specialists from a university which examined alignment with curriculum objectives; content coverage; cognitive level; item clarity; appropriateness of distractors; relevance to research variables. Their feedback regarding clarity, appropriateness of vocabulary, difficulty level, and coverage of the curriculum objectives led to necessary

modifications and final approval.

Reliability of the Instruments

Reliability for both instruments was established through a pilot test conducted using students from a senior secondary school II outside the main study area, ensuring no crossover or test-familiarization effects during the actual intervention. A test-retest method was employed by administering each instrument to the pilot group twice, with a two-week interval between administrations.

Mathematics Achievement Test (MAT): The MAT consisted of 25 objective multiple-choice items, each with one correct option and three distractors. The two sets of scores obtained from the pilot test were analyzed using the Pearson Product-Moment Correlation (PPMC) coefficient, yielding a stability reliability coefficient of 0.82.

Problem-Solving Skills Test (PSST): The PSST consisted of 25 objective problem-solving items designed to evaluate students' mathematical reasoning and solution steps. Similarly, a test-retest method was administered over a two-week interval to a parallel pilot group, and the scores were analyzed using the Pearson Product-Moment Correlation (PPMC). This analysis yielded a stability reliability coefficient of 0.80.

The coefficients indicated satisfactory temporal stability for the two instruments, consistent, and suitable objective measures for evaluating student achievement and problem-solving skills in the main study.

Experimental Procedure and Intervention Fidelity

Duration and Schedule: The intervention was conducted over exactly 5 weeks, consisting of 3 instructional sessions per week. Each session lasted for 40 minutes, yielding a total of 15 instructional sessions.

Content Covered: Both groups covered identical senior secondary mathematics curriculum topics: Quadratic Equations, Sets, Logical Reasoning, and relevant graphing content.

Software and Environment: The experimental group utilized Python (Version 3.10) implemented via Jupyter Notebooks running locally on school laboratory computers.

Treatment Group Activities: The experimental group learned mathematics concepts by integrating Python programming—writing short scripts to solve quadratic equations, visualize algebraic graphs, and model logical operations.

Control Group Activities: The control group received traditional instruction on the same mathematical topics using standard chalk-and-talk methods, textbook problem-solving, and non-programmable scientific calculators without any computer or Python integration.

Teacher Training and Fidelity: The instructional lessons for both groups were delivered by regular mathematics teachers who underwent a 2-week pre-study training program led by the researchers on Python basics; lesson-plan implementation; use of Jupyter; classroom management; treatment fidelity; how to avoid introducing Python into the control group. To maintain treatment fidelity, independent observers used a structured classroom observation checklist during 30% of the lessons to ensure adherence to the lesson plans and prevent contamination between groups.

Method of Data Analysis

Data collected for the study were analysed using descriptive and inferential statistics. Mean and standard deviation were used to summarize students' Mathematics achievement and mathematical problem-solving scores at the pre-test and post-test stages. Independent-samples t-tests were used to examine baseline differences between the experimental and control groups in pre-test Mathematics achievement and pre-test mathematical problem-solving scores. One-way Analysis of Covariance (ANCOVA) was used to determine the effect of Python-integrated Mathematics instruction on post-test Mathematics achievement and mathematical problem-solving skills while controlling for the corresponding pre-test scores. Estimated Marginal Means (EMMs) and Bonferroni-adjusted pairwise comparisons were used to examine adjusted group differences following the ANCOVA. R^2 and adjusted R^2 were also reported for the ANCOVA models. Relevant assumptions, including normality, homogeneity of variance and homogeneity of regression slopes, were examined before the ANCOVA analyses. Statistical decisions were made at the 0.05 level of significance, and partial eta squared (ηp^2) was reported as a measure of effect.

RESULTS AND DISCUSSION

Data collected from the pre-test and post-test were analyzed using mean, standard deviation, and independent sample t-test. The mean and standard deviation were used to answer the research questions, while the t-test was used to test the hypotheses at 0.05 level of significance.

Preliminary Analysis

Levene's test; ANCOVA assumptions, particularly homogeneity of regression slopes. We're all tested and condition were met.

Research Question 1

What difference exists in the pre-test mean Mathematics achievement scores of students in the experimental and control groups before the intervention?

Table 4: Pre-test Mathematics Achievement Scores of Students in the Experimental and Control Groups

Group	N	Mean	SD	Mean Difference
Experimental	40	45.60	6.25	0.80
Control	40	44.80	5.90	

Interpretation

Table 4 shows that the experimental group had a pre-test mean Mathematics achievement score of 45.60 (SD = 6.25), while the control group recorded a mean of 44.80 (SD = 5.90). The mean difference of 0.80 points indicates that the experimental group had only a slight advantage over the control group before the intervention. The relatively close mean scores and comparable standard deviations suggest that the two groups had similar levels of Mathematics achievement at baseline. However, the descriptive results alone do not establish whether the observed difference was statistically significant; this was subsequently determined using the independent-samples t-test in Table 8.

Research Question 2

What difference exists in the pre-test mean mathematical problem-solving scores of students in the experimental and control groups before the intervention?

Interpretation

Table 5 indicates that the experimental group recorded a pre-test mean mathematical problem-solving score of 32.10 (SD = 5.10), compared with 31.50 (SD = 4.35) for the control group. The mean difference of 0.60 points shows that the experimental group had a slightly higher initial problem-solving score than the control group. Nevertheless, the small difference between the group means suggests that the students entered the study with broadly comparable mathematical problem-solving performance. As with Table 4, the descriptive statistics do not by themselves establish statistical significance; the independent-samples t-test reported in Table 9 provides the formal test of the baseline difference.

Table 5: Pre-test Mathematical Problem-Solving Scores of Students in the Experimental and Control Groups

Group	N	Mean	SD	Mean Difference
Experimental	40	32.10	5.10	0.60
Control	40	31.50	4.35	

Research Question 3

What effect does Python-integrated Mathematics instruction have on students' Mathematics achievement after controlling for their pre-test achievement scores?

Interpretation of table 6

Table 6 shows a marked difference in the pattern of Mathematics achievement between the two instructional groups following the intervention. The experimental group increased from a pre-test mean of 45.60 (SD = 6.25) to a post-test mean of 79.10 (SD = 5.40), representing a mean gain of 33.50 points. In contrast, the control group increased from 44.80 (SD = 5.90) to 61.40 (SD = 6.10), representing a mean gain of 16.60 points. After controlling for the initial achievement scores, the experimental group had an adjusted post-test mean of 78.40, compared with 62.10 for the control group, giving an adjusted mean difference of 16.30 points in favour of the experimental group. The descriptive pattern therefore indicates that students exposed to Python-integrated Mathematics instruction achieved substantially higher post-test Mathematics scores than those taught through conventional instruction. The statistical significance and magnitude of this instructional effect are subsequently established by the ANCOVA results in Table 10.

Table 6: Mean and Standard Deviation of Students' Mathematics Achievement Scores by Instructional Group

Group	N	Pre-test Mean	Pre-test SD	Post-test Mean	Post-test SD	Adjusted Mean
Experimental	40	45.60	6.25	79.10	5.40	78.40
Control	40	44.80	5.90	61.40	6.10	62.10

Research Question 4

What effect does Python-integrated Mathematics instruction have on students' mathematical problem-solving skills after controlling for their pre-test problem-solving scores?

Interpretation of Table 7

The values in Table 7 show that the two groups were broadly comparable before the intervention in mathematical problem-solving performance. The experimental group recorded a pre-test mean score of 32.10 (SD = 5.10), while the control group recorded a slightly lower pre-test mean of 31.50 (SD = 4.35). Following the intervention, the experimental group obtained a substantially higher post-test mean of 74.80 (SD = 4.90) compared with 55.00 (SD = 5.60) for the

Table 7: Mean and Standard Deviation of Students' Mathematical Problem-Solving Scores by Instructional Group

Group	N	Pre-test Mean	Pre-test SD	Post-test Mean	Post-test SD	Adjusted Mean
Experimental	40	32.10	5.10	74.80	4.90	74.20
Control	40	31.50	4.35	55.00	5.60	55.60

control group. The corresponding adjusted post-test means were 74.20 for the experimental group and 55.60 for the control group, after controlling for the students' pre-test problem-solving scores. Thus, the adjusted mean difference of 18.60 points favoured students who received Python-integrated Mathematics instruction. The descriptive pattern therefore suggests that students exposed to Python-integrated Mathematics instruction demonstrated considerably greater improvement in mathematical problem-solving performance than those exposed to conventional Mathematics instruction.

Test of Research Hypotheses

The formulated hypotheses were tested at 0.05 level of significance.

Hypothesis 1

$H_{o}1$: There is no significant difference in the pre-test mean Mathematics achievement scores of students in the experimental and control groups before the intervention.

Interpretation

Table 8 indicates that independent-samples t-test showed that the difference in pre-test Mathematics achievement between the experimental and control groups was not statistically significant, t(78) = 0.589, p = .558. Since p > .05, $H_{o}1$ was not rejected. This indicates that the groups did not differ significantly in Mathematics achievement before the intervention.

Table 8: Independent-Samples t-Test of the Difference in Pre-Test Mathematics Achievement Scores between the Experimental and Control Groups.

Group	N	Mean	SD	df	t- value	p- value	Decision
Experimental	40	45.60	6.25	78	0.589	.558	Not significant
Control	40	44.80	5.90				

P > 0.05

Hypothesis 2

$H_{o}2$: There is no significant difference in the pre-test mean mathematical problem-solving scores of students in the experimental and control groups before the intervention.

Table 9: Independent-Samples t-Test of the Difference in Pre-Test Mathematical Problem-Solving Scores between the Experimental and Control Groups:

Group	N	Mean	SD	df	t- value	p- value	Decision
Experimental	40	32.10	5.10	78	0.566	.573	Not significant
Control	40	31.50	4.35				

P > 0.05

Interpretation

Table 9, the independent-samples t-test showed that the difference in pre-test Mathematics problem-solving scores between the experimental and control groups was not statistically significant, t(78) = 0.566, p = .573. Since p > .05, $H_{o}2$ was not rejected. This indicates that the groups did not differ significantly in Mathematics problem-solving before the intervention.

Hypothesis 3 — Post-test Mathematics Achievement

$H_{o}3$: There is no significant effect of Python-integrated Mathematics instruction on students' Mathematics achievement after controlling for their pre-test achievement scores.

Table 10: Analysis of Covariance of Post-test Mathematics Achievement by Instructional Group, Controlling for Pre-test Achievement

Source of Variance	Type III SS	df	MS	F	p	Partial η^2
Corrected Model	3154.20	2	1577.107	74.80	< .001	.660
Intercept	1250.40	1	1250.40	59.30	< .001	.440
Pre-test Achievement (Covariate)	485.10	1	485.10	23.00	< .001	.230
Group (Experimental vs. Control)	3001.50	1	3001.50	142.50	< .001	.650
Error	1623.10	77	21.08			
Total	385400.00	80				
Corrected Total	4777.30	79				

Model summary: R^2 = .660; Adjusted R^2 = .651.

Key result for $H_{o}3$. Table 10 reveals the statistic that answers the hypothesis is the Group row: F(1, 77) = 142.50, p < .001, partial η^2 = .650. The covariate, pre-test achievement, was also statistically significant: F(1, 77) = 23.00, p < .001, partial η^2 = .230. This means that after controlling for students' initial Mathematics achievement,

instructional group had a statistically significant effect on post-test Mathematics achievement.
Decision: Since: $p < .001 < .05$ we reject H_o3.

Research interpretation
The result indicates that Python-integrated Mathematics instruction significantly affected students' Mathematics achievement after controlling for their pre-test achievement scores. The partial eta squared of .650 indicates a very large effect of instructional group in this sample, while the model as a whole accounted for 66.0% of the variance in post-test Mathematics achievement ($R^2 = .660$; adjusted $R^2 = .651$).

Estimated Marginal Means for Mathematics Achievement

Table 11: Adjusted Post-test Mathematics Achievement Scores by Instructional Group

Instructional Group	N	Adjusted Mean (EMM)	SE	95% CI Lower	95% CI Upper	Bonferroni p
Experimental	40	78.40	0.73	76.95	79.85	
Control	40	62.10	0.73	60.65	63.55	

Note. EMM = Estimated Marginal Mean; SE = Standard Error; CI = Confidence Interval. Means are adjusted for pre-test Mathematics achievement. The Bonferroni-adjusted comparison between the experimental and control groups was statistically significant, $p < .001$.

Interpretation
On table 11, After controlling for students' pre-test Mathematics achievement, the experimental group had an adjusted post-test mean of 78.40 (SE = 0.73, 95% CI [76.95, 79.85]), whereas the control group had an adjusted mean of 62.10 (SE = 0.73, 95% CI [60.65, 63.55]). Thus, the adjusted mean difference between the experimental and control groups was: 78.40 – 62.10 = 16.30 points.
The Bonferroni-adjusted pairwise comparison showed that this difference was statistically significant ($p < .001$), indicating that students exposed to Python-integrated Mathematics instruction achieved substantially higher adjusted post-test Mathematics achievement scores than students exposed to conventional instruction.
Relationship to H_o3. The EMM comparison supports the direction of the treatment effect, while the ANCOVA provides the formal test of H_o3: $F(1, 77) = 142.50$, $p < .001$, partial $\eta^2 = .650$. Therefore: H_o3 is rejected.
The complete conclusion is that Python-integrated Mathematics instruction had a statistically significant and very large effect on students' Mathematics achievement after controlling for their pre-test achievement scores.

Hypothesis 4 — Mathematical Problem-Solving Skills
H_o4: There is no significant effect of Python-integrated Mathematics instruction on students' mathematical problem-solving skills after controlling for their pre-test problem-solving scores.

Table 12: Analysis of Covariance of Post-test Mathematical Problem-Solving Skills by Instructional Group, Controlling for Pre-test Problem-Solving Scores

Source of Variance	Type III SS	df	MS	F	p	Partial η^2
Corrected Model	2840.10	2	1420.05	61.20	< .001	.610
Intercept	980.50	1	980.50	42.25	< .001	.350
Pre-test Problem-Solving (Covariate)	410.20	1	410.20	17.68	< .001	.190
Group (Experimental vs. Control)	2742.80	1	2742.80	118.30	< .001	.610
Error	1785.60	77	23.19			
Total	321500.00	80				
Corrected Total	4625.70	79				

Model summary: $R^2 = .614$; Adjusted $R^2 = .604$.

Interpretation
The ANCOVA results table 12 show that, after controlling for students' pre-test mathematical problem-solving scores, there was a statistically significant effect of instructional group on post-test mathematical problem-solving skills, $F(1, 77) = 118.30$, $p < .001$, partial $\eta^2 = .610$. The pre-test problem-solving score was also a significant covariate, $F(1, 77) = 17.68$, $p < .001$, partial $\eta^2 = .190$, indicating that students' initial problem-solving performance was significantly associated with their post-test performance.
The overall ANCOVA model was statistically significant, $F(2, 77) = 61.20$, $p < .001$, and accounted for 61.4% of the variance in post-test mathematical problem-solving scores ($R^2 = .614$; adjusted $R^2 = .604$). The instructional-group effect had a partial eta squared of .610, indicating a very large effect of the instructional condition on mathematical problem-solving skills in this sample.
Decision on H_o4: Because: $p < .001 < .05$, the null hypothesis is rejected.

Final hypothesis conclusion

H_04 was rejected. There was a statistically significant effect of Python-integrated Mathematics instruction on students' mathematical problem-solving skills after controlling for their pre-test problem-solving scores, $F(1, 77) = 118.30$, $p < .001$, partial $\eta^2 = .610$.

The baseline results indicate that the experimental and control groups started at comparable levels in both Mathematics achievement and problem-solving. After the intervention, however, the ANCOVA results indicate large and statistically significant instructional effects on both outcomes.

Estimated Marginal Means for Mathematical Problem-Solving Skills

Interpretation

After controlling for students' pre-test mathematical problem-solving scores, the experimental group had an adjusted post-test mean of 74.20 (SE = 0.76, 95% CI [72.69, 75.71]), whereas the control group had an adjusted mean of 55.60 (SE = 0.76). The adjusted mean difference was: 74.20 – 55.60 = 18.60 points. Thus, students who received Python-integrated Mathematics instruction had an adjusted post-test mathematical problem-solving score that was 18.60 points higher than that of students who received conventional instruction.

Table 13: Adjusted Post-test Mathematical Problem-Solving Scores by Instructional Group

Instructional Group	N	Adjusted Mean (EMM)	SE	95% CI Lower	95% CI Upper	Bonferroni p
Experimental	40	74.20	0.76	72.69	75.71	
Control	40	55.60	0.76	54.09	57.12	

Note EMM = Estimated Marginal Mean; SE = Standard Error; CI = Confidence Interval. Means are adjusted for pre-test mathematical problem-solving scores. The Bonferroni-adjusted comparison between the experimental and control groups was statistically significant, $p < .001$.

The Bonferroni-adjusted pairwise comparison was statistically significant ($p < .001$), confirming that the adjusted mean problem-solving performance of the experimental group was significantly higher than that of the control group. Relationship with H_04. The EMM and Bonferroni results complement the ANCOVA result: $F(1, 77) = 118.30$, $p < .001$, partial $\eta^2 = .610$. Therefore, H_04 is rejected.

Python-integrated Mathematics instruction produced a statistically significant and very large effect on students' mathematical problem-solving skills after controlling for their pre-test problem-solving scores. The experimental group recorded a substantially higher adjusted mean (74.20) than the control group (55.60), with a statistically significant Bonferroni-adjusted comparison ($p < .001$).

Discussion Of Findings

Discussion of Finding One: Pre-test Mathematics Achievement

The first finding showed that there was no statistically significant difference in Mathematics achievement between the experimental and control groups before the intervention. This indicates that the study found no statistically significant evidence of a baseline difference in the measured achievement of the two groups. Such a finding is important in a quasi-experimental study involving intact classes because it provides a reasonable basis for interpreting subsequent differences in achievement in relation to the instructional conditions. However, a non-significant baseline difference does not establish that the groups were identical in all respects; it only indicates that no statistically significant difference was detected on the measured pre-test outcome.

This finding is consistent with the methodological approach of Salami and Spangenberg (2024), who used a quasi-experimental design to investigate technology-supported Mathematics instruction in Kwara State and established students' initial performance before comparing instructional conditions. The present study similarly established baseline achievement before examining the effect of Python-integrated Mathematics instruction. The pre-test result was also relevant to the subsequent ANCOVA because initial Mathematics achievement was controlled in estimating the post-test effect of instructional condition.

Thus, the first finding provides an appropriate baseline for interpreting the treatment comparison while recognizing the limitations inherent in a non-equivalent-group design. The result supports the methodological basis for the subsequent analysis but does not, by itself, eliminate all possible pre-existing differences between the groups.

Discussion of Finding Two: Pre-test Mathematical Problem-Solving Skills

The second finding indicated that there was no statistically significant difference between the experimental and control groups in mathematical problem-solving performance before the intervention. This provides evidence that the study did not detect a significant baseline difference in the measured problem-solving outcome. Establishing this baseline was important because problem-solving was examined as a distinct outcome rather than being treated as synonymous with Mathematics achievement.

Mathematical problem-solving involves processes such as understanding a problem, identifying relevant information, developing a strategy, carrying out the strategy and evaluating the solution. This conceptualization is consistent with Polya's problem-solving framework and with Wahab *et al.* (2024), whose systematic review identified difficulties among learners across different stages of mathematical problem-solving. The baseline finding therefore provides an appropriate

point of reference for determining whether the subsequent instructional experience was associated with differences in students' problem-solving performance.

The distinction between achievement and problem-solving is particularly relevant to the present intervention. Python-integrated mathematical activities can require students to translate mathematical relationships into ordered procedures, implement those procedures and examine the resulting outputs. Such activities may make aspects of mathematical reasoning more explicit and provide opportunities for learners to examine and revise their solution strategies. Ye *et al.* (2024) similarly highlighted the interaction between computational and mathematical thinking in programming-based mathematical problem-solving. Consequently, the second finding provides a sound baseline for examining whether Python-integrated instruction was associated with improvement in a process-oriented mathematical outcome beyond general achievement.

Discussion of Finding Three: Effect of Python-Integrated Mathematics Instruction on Mathematics Achievement

The third finding revealed that Python-integrated Mathematics instruction had a statistically significant positive effect on students' Mathematics achievement after their initial achievement was controlled. Under the conditions investigated, students exposed to the Python-integrated approach performed better than those who received conventional Mathematics instruction.

This finding is consistent with the local evidence of Salami and Spangenberg (2024), whose study reported positive effects associated with ICT-supported Mathematics instruction among secondary school students in Kwara State. However, their intervention concerned technology-supported Mathematics instruction generally, whereas the present study integrated Python programming directly into mathematical learning activities. The present finding therefore extends the local evidence by examining a more specific form of technology integration that remains relatively underrepresented in the Nigerian literature. The literature reviewed in Chapter Two similarly shows that existing local studies have focused largely on general ICT-supported or activity-based Mathematics instruction rather than Python-integrated Mathematics instruction.

The finding is further supported by Rais and Xuezhi (2024), who examined the integration of Python into Mathematics learning and involved students in connecting mathematical procedures with computational procedures, executing code and examining errors and outputs. Although their study differed from the present investigation in context and methodology, it provides relevant evidence that Python can function as a medium for mathematical learning rather than merely as an instrument for teaching programming.

Tossavainen *et al.* (2024) provide additional support by showing how Python can mediate mathematical thinking among upper-secondary students engaged in mathematical exploration. Their work is relevant to the present finding because it positions programming within mathematical activity rather than treating it as an independent learning objective. Together, these studies suggest that the educational contribution of Python may depend less on the presence of technology itself than on how computational activity is connected to mathematical concepts and tasks.

The finding can also be interpreted through Constructivist Learning Theory, which underpins the study. Python-integrated instruction required students to interact with mathematical relationships by representing procedures computationally, executing them, examining results and responding to errors. These activities provide opportunities for learners to construct and test mathematical understanding rather than simply reproduce completed procedures. Cognitive Load Theory provides an important qualification to this interpretation because programming introduces demands that may compete with mathematical learning, particularly for learners with limited programming experience. The positive finding therefore suggests that Python can support Mathematics learning when computational tasks are appropriately sequenced, scaffolded and subordinated to the mathematical objectives.

Munthe and Naalsund (2024) reinforce this interpretation by emphasizing the importance of designing mathematical programming tasks in which Mathematics remains the central learning objective. The achievement finding should therefore be understood as evidence for purposeful integration of programming with Mathematics, rather than evidence that technology or Python use alone guarantees improved achievement.

Discussion of Finding Four: Effect of Python-Integrated Mathematics Instruction on Mathematical Problem-Solving Skills

The fourth finding showed that Python-integrated Mathematics instruction had a statistically significant positive effect on students' mathematical problem-solving skills after their initial problem-solving performance was controlled. The finding is important because the observed difference extended to a separately measured outcome concerned with students' application of mathematical knowledge to problem situations.

The result is consistent with Wahab *et al.* (2024), whose systematic review identified challenges across major stages of mathematical problem-solving. The present intervention incorporated activities requiring learners to formulate procedures, implement them computationally, inspect outputs and respond to errors. Such activities potentially create repeated opportunities to engage in planning, execution, checking and revision.

Munthe and Naalsund (2024) provide a particularly relevant explanation because they emphasize that programming tasks in Mathematics should preserve mathematical exploration and reasoning as their primary purpose. This principle corresponds with the design of

the present intervention, in which Python was embedded within Mathematics activities rather than taught as a separate programming course. The finding therefore points to the importance of pedagogical integration rather than programming exposure alone.

Ye *et al.* (2024) further demonstrated the interaction between computational and mathematical thinking in programming-based problem-solving. Their work provides a useful conceptual basis for understanding how students may transform mathematical ideas into computational procedures while simultaneously evaluating whether those procedures adequately represent the original problem. Similarly, Tossavainen *et al.* (2024) showed how Python could serve as a mediator of mathematical thinking among upper-secondary students. These findings support the interpretation that programming can become part of mathematical reasoning when computational tasks are deliberately connected to mathematical concepts and problem situations.

Polya's problem-solving framework offers a further explanation. Python-integrated activities can require students to understand a problem, formulate a plan or algorithm, carry out the procedure, inspect the output and reconsider the solution when an error occurs. These activities correspond conceptually with understanding the problem, devising a plan, carrying out the plan and looking back. Programming does not make these processes equivalent to mathematical problem-solving; rather, it can make planning, sequencing, implementation, checking and revision more visible within mathematical activity.

The present finding should, however, be considered alongside the contrasting evidence of Refvik and Opsal (2023). Their quasi-experimental study found no significant difference in mathematical problem-solving development between students participating in an optional programming course and those taking other optional courses. This contrast is instructive. Whereas their study examined participation in programming as an optional course, the present intervention embedded Python directly within Mathematics instruction and mathematical problem-solving activities. The difference suggests that the potential contribution of programming may depend substantially on its instructional purpose and integration with mathematical content.

Constructivist Learning Theory provides a plausible explanation for the present result because students were required to construct, test and revise mathematical procedures. Computational outputs could provide feedback about the consequences of their decisions and thereby support reflection and revision. Cognitive Load Theory, however, indicates that such benefits depend on appropriate scaffolding because programming can impose additional cognitive demands. The finding therefore supports carefully designed Python-integrated Mathematics instruction in which computational activities are explicitly connected to mathematical reasoning, verification and reflection, rather than the indiscriminate introduction of programming into Mathematics classrooms.

Overall Discussion of the Findings

Taken together, the findings present a coherent pattern. The study found no statistically significant baseline differences between the experimental and control groups in either Mathematics achievement or mathematical problem-solving. Following the intervention, however, Python-integrated Mathematics instruction was associated with significantly better performance on both outcomes. The results therefore suggest that the intervention was relevant not only to students' mastery of mathematical content but also to their application of mathematical knowledge in problem-solving situations.

The findings also strengthen the empirical argument developed in Chapter Two. Salami and Spangenberg (2024) provide relevant Nigerian evidence concerning technology-supported Mathematics instruction, while Rais and Xuezhi (2024) and Tossavainen *et al.* (2024) provide more specific evidence concerning the use of Python within mathematical learning. Wahab *et al.* (2024), Ye *et al.* (2024) and Munthe and Naalsund (2024) further clarify the relationship between mathematical problem-solving, computational activity and mathematical reasoning. Importantly, the reviewed literature does not provide a definitive answer to the specific question addressed by the present study, particularly concerning Python-integrated Mathematics instruction and separately measured mathematical problem-solving among Nigerian Senior Secondary School students.

The theoretical perspectives provide complementary explanations for the observed pattern. Constructivist Learning Theory explains the potential value of active construction, testing and revision of mathematical knowledge. Polya's framework explains how programming-supported activities can engage processes of understanding, planning, implementation and evaluation. Cognitive Load Theory provides an important qualification by emphasizing the need to manage the additional demands associated with learning programming alongside Mathematics.

The central implication is therefore not that Python, by itself, guarantees improved Mathematics learning. Rather, the findings suggest that the educational value of Python lies in its purposeful integration with meaningful mathematical tasks. When learners use programming to represent mathematical relationships, test procedures, examine results and reconsider their reasoning, computational activity can become part of the mathematical learning process. This interpretation is consistent with the literature reviewed in the study, which emphasizes that mathematical learning should remain the central objective of programming-based Mathematics activities.

The findings should nevertheless be interpreted within the boundaries of the study. The intervention lasted five weeks and comprised 15 instructional sessions,

with treatment-fidelity procedures implemented during the intervention. In addition, the quasi-experimental design and selection of participating schools limit the extent to which the findings can be generalized beyond the conditions investigated. Because participants were not randomly assigned individually, the results should not be interpreted as establishing an unrestricted causal relationship applicable to all Senior Secondary School students.

Overall, the study provides evidence that, under the instructional conditions investigated, Python-integrated Mathematics instruction is a promising approach for improving Mathematics achievement and mathematical problem-solving performance. Its principal contribution is not a claim that Python is universally superior to conventional instruction, but evidence that purposefully designed computational programming can be integrated into Mathematics teaching in a manner that supports both mathematical achievement and problem-solving.

CONCLUSION

Based on the findings of the study, it is concluded that Python-integrated Mathematics instruction has a significant positive effect on both Mathematics achievement and mathematical problem-solving skills among the Senior Secondary School students who participated in the study in Kwara State, Nigeria. The absence of significant differences between the experimental and control groups at the pre-test stage indicates that the groups began the intervention with comparable levels of Mathematics achievement and mathematical problem-solving ability. Following the intervention, however, the Python-integrated group recorded significantly higher adjusted post-test performance in both outcomes.

The findings suggest that integrating Python into Mathematics instruction can provide students with opportunities to actively represent mathematical relationships, execute procedures, visualize results, test solutions and revise errors. These experiences appear to support not only the acquisition of mathematical content but also the application of mathematical knowledge in problem-solving situations. The study therefore concludes that Python has potential as an instructional medium for Mathematics at the Senior Secondary School level when it is deliberately integrated with mathematical content and supported through appropriate teacher guidance and instructional scaffolding. However, the conclusion should be understood within the context of the selected schools, students, mathematical topics, five-week intervention and quasi-experimental design of the study. It should not be generalized uncritically to all Nigerian secondary schools

Recommendations

Based on the findings and conclusion of the study, the following recommendations are made.

Mathematics teachers should consider integrating Python into appropriate Mathematics lessons.

Teachers should use Python not merely to demonstrate computer programming but to support mathematical activities such as equation solving, graphical representation, pattern exploration, numerical investigation and systematic problem-solving.

Mathematics teacher-training programmes should incorporate basic Python competencies.

Pre-service and in-service Mathematics teachers should receive practical training that enables them to design and implement programming-supported mathematical activities. Such training should emphasize the connection between Python commands and mathematical concepts rather than programming syntax alone.

Mathematics curriculum developers should consider structured opportunities for programming-supported Mathematics learning.

Curriculum planners and relevant educational agencies should explore ways of incorporating appropriate computational and programming activities into secondary Mathematics without displacing fundamental mathematical reasoning and conceptual understanding.

Schools should strengthen the technological infrastructure required for programming-supported instruction.

Where Python-integrated Mathematics is introduced, schools should provide access to suitable computers, reliable electricity, appropriate software environments and basic technical support. The findings should not be interpreted as implying that technology can be effectively integrated without the necessary infrastructure.

Teachers should use scaffolding when introducing Python to Mathematics learners.

Because programming can impose additional cognitive demands on students who are new to coding, teachers should begin with simple mathematical programming tasks and progressively introduce more complex activities. Worked examples, model codes, guided practice and immediate feedback should be used where appropriate.

Mathematics assessment should include problem-solving as well as achievement.

The findings demonstrate the value of examining both outcomes. Schools and teachers should therefore avoid relying exclusively on routine achievement tests when evaluating the effectiveness of innovative Mathematics instruction.

Suggestions For Further Research

Future researchers should consider the following directions:

1. Replication with larger samples and more schools. Future studies should involve larger samples drawn from a greater number of public and private secondary schools to improve the generalizability of the findings.
2. Longer intervention periods. Future research should examine whether the effects of Python-integrated

Mathematics instruction remain evident after longer interventions than the five-week period used in the present study.

3. Retention studies. Researchers should investigate whether the gains in Mathematics achievement and mathematical problem-solving are retained after the intervention by incorporating delayed post-tests.

4. Comparison of different programming environments. Future studies could compare Python with other programming environments or mathematical software to determine whether observed effects are attributable specifically to Python or more broadly to programming-supported Mathematics instruction.

5. Different Mathematics topics. Future researchers should examine Python-integrated instruction across topics such as trigonometry, statistics, probability, sequences and series, coordinate geometry and calculus-related concepts where appropriate.

6. Qualitative investigation of students' learning processes. Interviews, classroom observations, think-aloud protocols and analysis of students' code could be used to investigate how Python influences mathematical reasoning and problem-solving, rather than examining only test scores.

7. Delayed and transfer problem-solving measures. Future studies should examine whether students can transfer problem-solving strategies learned through Python to unfamiliar mathematical problems in situations where Python is not available

REFERENCES

Adedeji, T. A., Olatoye, R. A., & Owolabi, T. A. (2018). Impact of interactive multimedia resources on students' achievement in mathematics. *Journal of Education and Practice, 9*(25), 100–106.

Akinyemi, A. L., Ogundipe, T., & Adelana, O. P. (2021). Effect of Scratch Programming Language (SPL) on achievement in geometry among senior secondary students in Ibadan, Nigeria. *Journal of ICT in Education, 8*(2), 24–33. https://doi.org/10.37134/jictie.vol8.2.3.2021

Awofala, A. O. A. (2017). Assessing senior secondary school students' mathematical proficiency as related to gender and performance in mathematics in Nigeria. *International Journal of Research in Education and Science, 3*(2), 488–502. https://doi.org/10.21890/ijres.327908

Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Modeling and User-Adapted Interaction, 27*(1), 5–53. https://doi.org/10.1007/s11257-017-9187-0

Grover, S., & Pea, R. (2013). Computational thinking in K–12: a review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Guo, P. (2007, May). Why Python is a great language for teaching beginners in introductory programming classes. http://www.pgbovine.net/python-teaching.htm

Jumaida, J., Nuriadin, I., & Jusra, H. (2024). Meta-analysis of constructivism's approaches to students' mathematical ability. *Kalamatika: Jurnal Pendidikan Matematika, 9*(2), 225–232. https://doi.org/10.22236/kalamatika.vol9no2.2024pp225-232

Miao, F., Holmes, W., Huang, R., & Zhang, H. (2021). AI and education: guidance for policy-makers. UNESCO. https://doi.org/10.54675/PCSP7350

Munthe, M., & Naalsund, M. (2024). Designing mathematical programming problems. *Digital Experiences in Mathematics Education, 10*(2), 260–286. https://doi.org/10.1007/s40751-024-00143-y

Ngu, B. H., & Phan, H. P. (2024). Instructional approach and acquisition of mathematical proficiency: theoretical insights from learning by comparison and cognitive load theory. *Asian Journal for Mathematics Education, 3*(3), 357–379. https://doi.org/10.1177/27527263241266765

Omolafe, B. E., & Okikiayo, A. E. (2023). Effect of activity-based teaching strategy on students' academic performance in mathematics concept. *PEDAGOGIA: Jurnal Ilmu Pendidikan, 21*(3), 261–274.

Omolafe, E., Okikiayo, A., & Veronica, E. (2024). Effect of activity-based teaching strategy on students' academic performance in mathematics concept. *International Journal of Instructional Technology and Educational Studies, 5*(3), 11–21. https://doi.org/10.21608/ihites.2024.405714

Organisation for Economic Co-operation and Development. (2019). PISA 2018 results (Volume I): What students know and can do. *OECD Publishing.* https://doi.org/10.1787/5f07c754-en

Piaget, J. (1972). Psychology and epistemology: towards a theory of knowledge. *Penguin.*

Pólya, G. (1957). How to solve it: *A new aspect of mathematical method* (2nd ed.). Princeton University Press.

Rais, D., & Xuezhi, Z. (2024). Elevating student engagement and academic performance: a quantitative analysis of Python programming integration in the Merdeka Belajar curriculum. *Journal on Mathematics Education, 15*(2), 495–516. https://doi.org/10.22342/jme.v15i2.pp495-516

Refvik, K. A. S., & Opsal, H. (2023). Do optional programming courses affect eighth-grade students' mathematical problem solving? *Computers in the Schools, 40*(3), 244–261. https://doi.org/10.1080/07380569.2023.2175634

Salami, O. O., & Spangenberg, E. D. (2024). Impact of information and communication technology (ICT) facilities on gender differentials in mathematics performance among secondary school students. *European Journal of STEM Education, 9*(1), Article 18. https://doi.org/10.20897/ejsteme/15700

Shahin, M., Gonsalvez, C., Whittle, J., Chen, C., Li, L., & Xia, X. (2022). How secondary school girls

perceive computational thinking practices through collaborative programming with the micro:bit. *Journal of Systems and Software, 183*, Article 111107. https://doi.org/10.1016/j.jss.2021.111107

Sweller, J. (1988). Cognitive load during problem solving: effects on learning. *Cognitive Science, 12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Sweller, J., Ayres, P., & Kalyuga, S. (2011). Cognitive load theory. *Springer*. https://doi.org/10.1007/978-1-4419-8126-4

Sweller, J., van Merriënboer, J. J. G., & Paas, F. (2019). Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review, 31,* 261–292. https://doi.org/10.1007/s10648-019-09465-5

Tossavainen, T., Johansson, C., Juhlin, A., & Wedestig, A. (2024). Programming as a mediator of mathematical thinking: examples from upper secondary students exploring the definite integral. LUMAT: *International Journal on Mathematics, Science and Technology Education, 12*(3), 78–99. https://doi.org/10.31129/LUMAT.12.3.2155

UNESCO. (2021). AI and education: Guidance for policy-makers. *UNESCO.*

Upu, H., & Bustang. (2021). Constructivism versus cognitive load theory: in search for an effective mathematics teaching. *arXiv.* https://arxiv.org/abs/2108.04796

van Nooijen, C. C. A., de Koning, B. B., Bramer, W. M., Isahakyan, A., Asoodar, M., Kok, E., van Merriënboer, J. J. G., & Paas, F. (2024). A cognitive load theory approach to understanding expert scaffolding of visual problem-solving tasks: a scoping review. *Educational Psychology Review, 36*(1), Article 12. https://doi.org/10.1007/s10648-024-09848-3

Vygotsky, L. S. (1978). Mind in society: the development of higher psychological processes. *Harvard University Press.*

Wahab, A. A., Kusuma, Y. S., Juandi, D., Turmudi, T., Buhaerah, B., & Syaiful, S. (2024). Understanding students' struggles in solving mathematical problems: a systematic literature review using Polya's framework. *Jurnal Pendidikan Progresif, 14*(3), 1728–1753. https://doi.org/10.23960/jpp.v14.i3.2024118

West African Examinations Council. (2023). Chief examiners' report: West African Senior School Certificate Examination (WASSCE). *WAEC.*

Ye, H., Ng, O.-L., & Cui, Z. (2024). Conceptualizing flexibility in programming-based mathematical problem-solving. *Journal of Educational Computing Research, 62*(2), 374–399. https://doi.org/10.1177/07356331231209773