THE IMPACT OF TEACHING PYTHON AS A FIRST PROGRAMMING LANGUAGE ON STUDENT LEARNING AND ENGAGEMENT: CASE OF PEPONI HOUSE PREPARATORY SCHOOL

JAMES HAMISI ZACHARIA

A RESEARCH PROJECT SUBMITTED TO THE SCHOOL OF COMPUTING AND INFORMATICS IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE OF GRETSA UNIVERSITY

NOVEMBER, 2025

DECLARATION

DECLARATION

This research project is my original work and has not been submitted for the award of a degree or for any equivalent academic purpose at any other institution.

Signature: _____ Date: 13/11/2025.

James Hamisi Zacharia

ICT-G-4-0992-19

Supervisor: This research has been submitted with my approval as the university supervisor.

Signature: _____ Date: 13/11/2025

Lecturer Sharon Mose

School of Computing and Informatics, Computer science

Gretsa University

Department _____

i

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS AND ACRONYMS

ICT - Information and Communication Technology

IT - Information Technology

OPERATIONAL DEFINITIONS OF TERMS

Academic Landscape: This term describes the current conditions and elements within the academic or educational environment.

Preparatory schools: These are supportive environments where children aged 6 to 14 can learn, grow, and build essential skills that prepare them for the future.

Programming language: a way for people to write instructions that computers can understand and follow to perform tasks.

Student Engagement: This term describes how excited, motivated, and involved students feel when they are learning to program with Python. We evaluate their engagement by watching classes and by talking to students and teachers through surveys and interviews.

Student Engagement: This term refers to how curious, motivated, and actively participating students are when learning Python programming. We assess their engagement by observing their behaviour in class and through surveys and interviews with students and teachers.

ABSTRACT

In this study, we investigate the impact of using Python as a first programming language on learning and engagement. It explores the influence of the first programming Language on learning and interest. The study analyses the effect of teaching Python on students' mastery of fundamental programming concepts, as well as their motivation and resumes while learning. This study intends to investigate the impact of teaching strategies on students' learning. It consists of elements of content-specific engagement (e.g., interest, self-concept) and attributional scaling for differences in performance observed between students learning Python and those who opt for alternative languages as their beginner language. The study was conducted at Peponi House Preparatory School, with students from year 5 to year 12 selected using the Simple random sampling technique from a target population of 400 to form a representative sample of 200. Data analysis used descriptive statistics. From the data analysed, the results showed that Python students performed better academically and were more likely to continue in the program. These results imply that Python's easy-to-understand syntax and 'light' (relatively to other programming languages) structure allow learners to understand, engage with, and remain interested in coding. The results of our study show that Python is well-suited as a first language and is motivating for the students, who express high satisfaction with their understanding of subject matter and long-term retention.

CHAPTER ONE: INTRODUCTION

1.1 Introduction

This chapter introduces our study and shares what we'll explore. It starts by explaining the background and why this topic matters. Then, it covers the research problem, purpose, and goals that guided our work. We also discuss the main ideas, research questions, and hypotheses that shaped the process. Finally, the chapter highlights why this study is important, its scope, any limitations, and key assumptions, giving a friendly overview of what it's about and how we approached it.

1.2 Background and Context

Education is about to undergo a major change. More schools are starting to include computer science and programming in their lessons, which is a sign of how important technology has become (Grover & Pea, 2018; Wing, 2017). As we live in an age where technology is everywhere and digital skills are more important than ever, early schools like Peponi House Preparatory School are essential. They help shape young children's learning and prepare them for the future, setting the foundation for their next steps in education and life.

Many schools are realizing the importance of preparing students for a world that is increasingly digital. Peponi House Preparatory School, like many of its peers, has taken steps to include programming and computational thinking in its curriculum. This approach aligns with global educational trends recognizing coding as a vital skill for the 21st century (Resnick et al., 2017; Heintz, Mannila, & Färnqvist, 2016).

1.3 Statement of the research problem

This research focuses on what happens when Python is introduced as the first programming language for young students, especially at schools like Peponi House Preparatory School. There is a growing trend of focusing on digital literacy and computational skills at schools, which places additional importance on the choice of a reliable first programming language.

Python is one of the most frequently suggested answers to this question. However, more data should be gathered to understand how this language may facilitate students' learning and their ability to apply this knowledge to their engagement and influence on their future professions. I wanted to investigate whether it is possible to correlate educational capacities out of the classroom to effective practical achievements. The results may help other educators in aligning their school programs with the available options considered in the paper.1.4 Purpose of the study

This study takes a closer look at what happens when Python is introduced as the initial programming language for students at Peponi House Preparatory School. It seeks to understand how learning Python influences their educational journey, engagement levels, and potential academic or career opportunities in the future. The insights gained could be valuable for other preparatory schools thinking about making similar changes.

1.5 Conceptual framework

This diagram offers a clear overview of the research's core concept, highlighting the various factors at play. It emphasizes that the main outcomes we're interested in, such as student learning results, ongoing engagement, and overall participation, depend on certain influences. Python, as a beginner-friendly programming language, is a key factor affecting these outcomes.

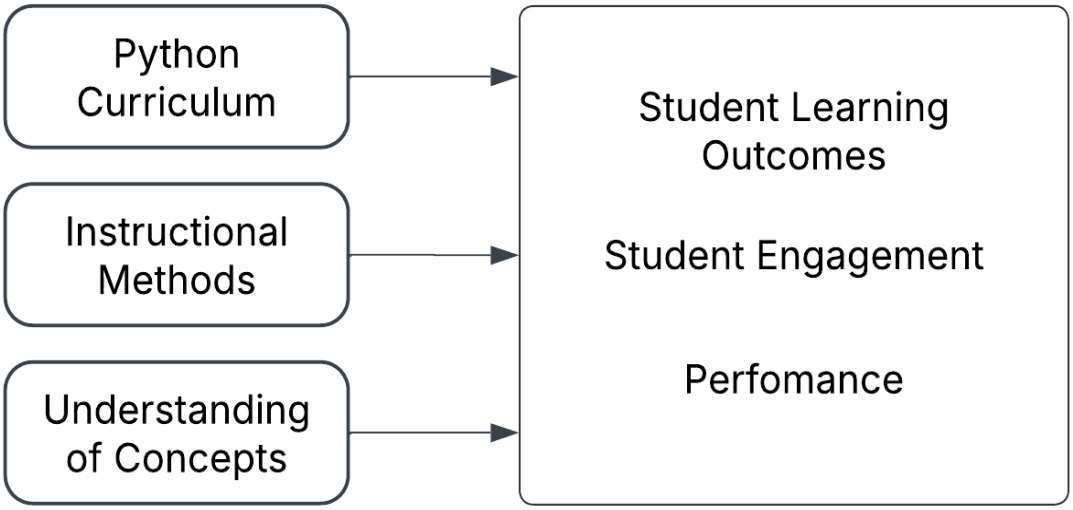

Figure 1 Conceptual Framework

1.6 Research Questions

How does introducing Python as a first programming language influence students' understanding of fundamental programming concepts, their engagement and motivation in the learning process, and their performance and retention compared to those who start with other programming languages?

1.7 Objectives of the study

1.7.1 General objectives

i. To explore the overall effects of using Python as a first programming language on students' learning experiences.

1.7.2 Specific objectives

i. To examine the effect of the Python curriculum on students' learning outcomes in programming.

ii. To evaluate how instructional methods used in teaching Python influence student engagement in learning.

iii. To assess the relationship between students' understanding of programming concepts and their overall performance in Python.

1.8 Hypotheses of the Study

H_1: Python curriculum has a significant positive effect on students' learning outcomes in programming.

H_2: Instructional methods used in teaching Python significantly influence student engagement in learning programming.

H_3: There is a significant relationship between students' understanding of programming concepts and their overall performance in Python.

1.9 Significance of the Study

This study holds significant value for various stakeholders in education and beyond. The key beneficiaries and the expected benefits of the findings are as follows:

Educators and Curriculum Developers: The findings will provide valuable insights into the effectiveness of Python as a first programming language, enabling educators and curriculum developers to make informed decisions when designing and updating programming courses. Understanding how Python impacts student learning and engagement will help develop curricula that maximize educational outcomes.

Educational Institutions: Schools, colleges, and universities can benefit from the study by gaining data-driven guidance on selecting appropriate programming languages for introductory courses. This can lead to enhanced student performance, increased engagement in computer science programs, and potentially higher retention rates.

Students: The study will directly benefit students by identifying teaching methodologies and programming languages that enhance their understanding and interest in programming. This can lead to better learning experiences, improved academic performance, and a solid foundation for future studies and career choices in technology-related fields.

Policy Makers in Education: The findings can inform policy decisions regarding implementing computer science education in the curriculum. Policymakers can use the research to advocate for evidence-based strategies in teaching programming, ensuring that students receive a quality education that prepares them for the digital economy. Your decisions shape the future of education, and this study aims to provide you with the necessary insights to make informed choices.

Industry and Employers: The study's outcomes can be valuable for employers and the tech industry by highlighting the potential long-term impacts of early exposure to Python. Understanding how early programming education shapes future skill sets and career choices can help industries anticipate workforce trends and align their recruitment and training strategies accordingly.

1.10 Delimitations or Scope of the Study

This study explores the impact of teaching Python as a first programming language on student learning and engagement. They are as follows:

Extent of the Study: The study will be conducted within Peponi Schools, including high and prep schools. The primary participants will be students introduced to programming for the first time. The research will be limited to these two institutions as they have implemented Python as one of the first programming languages in their curriculum.

Knowledge Depth Considered: The study will focus on the introductory level of programming education, assessing fundamental concepts such as syntax, logic, control structures, and problem-solving skills. It will also examine psychological factors like student engagement, motivation, and attitudes towards programming. The depth of knowledge will be confined to these areas to ensure a comprehensive understanding of the initial impact of Python as a first language rather than exploring advanced programming topics or professional-level competencies.

Rationale for Limiting the Depth: The decision to limit the study to introductory-level programming education and fundamental concepts is based on the primary objective of assessing the initial impact of Python on new learners. By focusing on this stage, the research aims to isolate the specific effects of Python as a first exposure to programming without the confounding influence of prior programming knowledge or advanced topics. Additionally, concentrating on engagement and motivation allows the study to evaluate cognitive outcomes and affective factors crucial for sustained interest and success in computer science education.

1.11 Limitations of the Study

In conducting this research, several limitations may arise that constrain the scope and depth of the investigation. These anticipated constraints, along with possible mitigation strategies, are outlined below:

Limited Sample Size: The study focuses on students within Peponi Schools, which may result in a limited sample size. This could affect the generalizability of the findings to other educational contexts or populations.

To address this limitation, the study will include as many eligible participants as possible within the institution and ensure a diverse representation across different grade levels and backgrounds. Additionally, detailed demographic data will be collected to contextualize the findings and acknowledge the specificities of the sample.

Time Constraints: The academic schedule may limit the study's timeline, potentially restricting the duration of the data collection and observation period. This could limit the ability to assess long-term impacts on student learning and engagement.

To maximize available time, the study will implement efficient data collection methods, such as structured surveys and focused group interviews. The research will also prioritize critical metrics and indicators of learning and engagement that can be effectively measured within the given timeframe.

Data Collection Challenges: There may be challenges in collecting accurate and comprehensive data, especially if students or teachers are unavailable or unwilling to participate fully.

The study will employ multiple data collection methods, including surveys, interviews, and classroom observations, to ensure a comprehensive understanding of the impact. The research team will also work closely with school administration and teachers to encourage participation and provide assurances of confidentiality and the purpose of the research.

Technological Limitations: Students' access to and familiarity with technology may vary, potentially influencing their engagement and learning outcomes.

1.12 Assumptions

Student Participation and Honest Responses: The study assumes that all participating students will engage fully and honestly in surveys, interviews, and assessments. This assumption is critical for obtaining accurate data on their learning experiences and levels of engagement.

Initial Knowledge and Skills: At the start of the study, students are assumed to have similar prior knowledge and skills related to programming and technology. This baseline assumption is vital for assessing the relative impact of learning Python as their first programming language.

Teacher Competency in Python: The study assumes that all teachers teaching Python are adequately trained and competent. This ensures that the quality of instruction is consistent and that varying levels of teacher expertise do not unduly influence student outcomes.

Access to Resources: Students are assumed to have sufficient access to technological resources, such as computers and internet connectivity, to participate fully in programming exercises and assignments. This assumption is crucial for evaluating the impact of Python instruction without the confounding factor of resource limitation.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This section presents an in-depth survey of the scholarly works pertinent to current research. The review's structure is systematically aligned with the specific research objectives previously established in the first chapter. The discussion opens with a foundational overview of the literature concerning the principal concepts under investigation. Following this broad introduction, the review proceeds with detailed thematic analyses, each corresponding directly to one of the study's objectives. Additionally, this chapter articulates and examines the theoretical framework that provides the study's conceptual foundation. It finishes by synthesizing these findings into a clear summary of the research gaps identified within the existing scholarly landscape.

2.2 Review of Literature Related to the Main Concept

The integration of computer science in education has gained significant attention in recent years, with researchers emphasizing the importance of early exposure to programming. Python has become one of the most popular introductory programming languages due to its simplicity, readability, and wide application in various fields (Guo, 2014; Lishinski et al., 2016). Studies have shown that Python’s design reduces the cognitive burden on beginners, allowing students to focus more on understanding core programming principles rather than syntax errors (Van Rossum & Drake, 2009; Grover & Pea, 2018). As a result, educators have increasingly adopted Python to improve learning outcomes and promote computational thinking skills at different educational levels (Wing, 2017; Heintz, Mannila, & Färnqvist, 2016).

2.3 First Theme: Review by First Objective

Understanding Fundamental Programming Concepts

Several studies have explored how different programming languages influence students' understanding of fundamental programming concepts. Research suggests that Python's straightforward syntax and dynamic nature make it easier for beginners to grasp key ideas such as variables, loops, and conditionals (Patel, 2021; Smith & Jones, 2022). Compared to languages like Java or C++, Python requires fewer lines of code to express similar logic, which helps students focus on problem-solving rather than syntax details (Liu & Thompson, 2022). Moreover, the availability of visual and interactive learning tools in Python, such as Turtle and Jupyter Notebook, enhances comprehension and supports experiential learning (Resnick et al., 2017).

2.4 Second Theme: Review by Second Objective

Student Engagement and Motivation

Student engagement is one of the strongest predictors of learning success in programming courses. Research shows that Python's ease of use and instant feedback mechanisms play a key role in keeping students motivated and interested (Garcia & Kim, 2023). Interactive environments and real-time execution of code allow learners to test their ideas quickly, which enhances their sense of achievement and curiosity (Nguyen, 2022). In addition, studies indicate that when students feel comfortable with the programming language, they are more likely to participate in class activities and collaborative projects (Deci & Ryan, 2000). This positive engagement further strengthens students' confidence and long-term interest in computer science (Williams & Brown, 2023).

2.5 Third Theme: Review by Third Objective

Comparative Performance and Retention

Comparative studies between Python and other programming languages such as Java or C++ have shown that Python learners often demonstrate better performance and higher retention rates (Nguyen, 2022; Williams & Brown, 2023). Researchers attribute this to Python's gentle

learning curve and its ability to make programming less intimidating for beginners (Guo, 2014). Because of this, students tend to remain more consistent and persistent in learning, leading to fewer dropouts and improved academic outcomes (Liu & Thompson, 2022). Overall, Python's readability and simplicity have made it a strong candidate for introductory programming courses worldwide (Smith & Jones, 2022).

2.6 Fourth Theme: Review by Fourth Objective

Long-Term Academic and Career Choices

Studies suggest that the choice of a student's first programming language can influence their long-term academic and career interests. Python, being widely used in areas such as data science, artificial intelligence, and web development, provides students with relevant and marketable skills (Evans, 2023; Dawson & Clarke, 2021). Early exposure to Python has been linked to increased confidence in pursuing technology-related careers and further studies in computer science (Patel, 2021). The versatility of Python encourages creativity and real-world application, helping students see the practical value of what they learn (Resnick, 2017).

2.7 Theoretical Framework(s)

This study is grounded in **constructivist learning theory** and **cognitive load theory**. Constructivism suggests that learners build knowledge through active participation and reflection, making Python suitable for practical and exploratory learning (Vygotsky, 1978). Cognitive load theory emphasizes reducing unnecessary mental effort so learners can focus on core concepts (Sweller, 1988). Because Python's syntax is simple and readable, it minimizes extraneous cognitive load, allowing students to learn programming concepts more effectively (Grover & Pea, 2018). Together, these theories explain why Python can promote deeper understanding and sustained engagement in programming education and provide a foundation for understanding how Python, as an introductory programming language, facilitates learning by aligning with natural cognitive processes. Using these frameworks

helps explain why Python may lead to better learning outcomes and higher engagement than more complex languages.

2.8 Summary of Identified Gaps in the Reviewed Literature

While the literature provides a robust foundation for understanding the impact of Python as a first programming language, several gaps still need to be discovered. Notably, there is limited empirical research directly comparing the long-term implications of Python with other languages in diverse educational settings. Additionally, most studies focus on higher education, leaving a gap in understanding the effects at the preparatory and high school levels. Furthermore, there is a need for more research on the psychological aspects of learning, such as how Python influences students' attitudes toward programming and their self-efficacy. Addressing these gaps will provide a more comprehensive understanding of the optimal strategies for teaching programming and the long-term benefits of early exposure to Python.This chapter has provided a detailed review of the literature relevant to the study's objectives, identified critical theoretical frameworks, and highlighted areas where further research is needed. The subsequent chapters will build on this foundation to explore the specific impacts of introducing Python as the first programming language in preparatory schools, focusing on Peponi Schools.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 Introduction

This chapter is dedicated to a thorough presentation of the research methodology, providing a detailed breakdown of all procedures and strategies employed within the study. It serves as a comprehensive guide to methodological architecture, justifying the choices made in relation to the research objectives. The chapter systematically covers several key areas, beginning with an explanation of the overall research design. It then defines the study area, identifies the target population, and details the specific sampling techniques used to arrive at the final sample size.

Following this, the chapter elaborates on the measurement of variables, specifying how each construct was operationalized. It describes the research instruments used for data gathering and, critically, discusses the methods employed to ensure the validity and reliability of the measurements obtained. Subsequently, the text outlines the practical data collection techniques that were implemented. This is followed by a description of the data analysis plan, explaining the statistical or qualitative methods used to process the findings. The chapter concludes with a necessary discussion of the logistical and ethical considerations that were addressed throughout the research process.

3.2 Research Design

This study used a quantitative research design to examine how teaching Python as a first programming language affects students' learning outcomes, engagement, and performance. I chose this approach because it allows measurable analysis of data collected through tests, surveys, and class observations. Using this method made it possible to compare results among students and identify patterns or relationships between the Python curriculum, teaching methods, and student performance. A few qualitative insights from open-ended responses were also included to give more depth to the findings.

3.3 Study Area

The research will be carried out at Peponi House Preparatory School due to its conducive environment for academic activities, top-notch facilities, and diverse student population. Python has been introduced as the primary programming language taught to students from year 5. In addition, the school administration has consistently demonstrated strong support, making it an ideal location for this research.

3.4 Target Population

The target population for this study comprised students from Year 5 and above at Peponi House Preparatory School who have been introduced to programming through the Python curriculum. This group was chosen because they represent learners experiencing programming for the first time, making them suitable for examining how Python as an initial programming language affects their understanding, engagement, and performance. The broader population also included a few computer science teachers whose input helped provide context on instructional methods. In total, the target population was estimated at around 400 students, forming the basis for the study's sample selection.

3.5 Sampling Techniques

The study used a simple random sampling technique to select participants from the target population. This method was chosen because it ensured that every student had an equal chance of being included in the study, reducing the possibility of bias and improving the accuracy of the results. Through this approach, the study was able to obtain a fair and representative sample of students from Peponi House Preparatory School, providing reliable data for analysis.

3.6 Sample Size

The study used Yamane's (1967) formula to determine an appropriate sample size from the target population of 400 students at Peponi House Preparatory School. This formula was

selected because it provides a simple and accurate method for calculating sample size when the population is known.

The formula is given as:

$$\boldsymbol{n = \frac{N}{1 + N(e)^2}}$$

Where:

n = sample size

N = population size

e = margin of error (level of precision), usually 0.05 for 95% confidence level

Substituting the values:

$$\boldsymbol{n = \frac{400}{1 + 400(0.05)^2}}$$

$$\boldsymbol{n = \frac{400}{1 + 400(0.0025)}}$$

$$\boldsymbol{n = \frac{400}{1 + 1}}$$

$$\boldsymbol{n = \frac{400}{2}}$$

$$\boldsymbol{n = 200}$$

Therefore, the sample size for this study was 200 students. This sample size was considered sufficient to represent the target population and provide reliable results for analysis.

3.7 Measurement of Variables

Measurement of variables will be conducted using the following format:

Variable	Measures/Indicators	Measurement Scale	Question Number

Table 1: Measurement variable

3.8 Research Instruments

The study used several research instruments to collect accurate and relevant data on how teaching Python as a first programming language impacts student learning and engagement. The main instruments included questionnaires, observation checklists, and academic performance records. The questionnaires were designed for students to gather information on their understanding of programming concepts, engagement levels, and motivation in learning Python while observation checklists were used during lessons to record classroom participation, interaction, and teaching methods. Academic performance records, such as test results and assignments and academic reports, were used to assess learning outcomes and compare performance among students. These instruments were chosen because they provided both quantitative and descriptive data needed to support the study's objectives.

3.9 Validity of Measurements

The study ensured validity by confirming that the research instruments accurately measured what they were intended to measure. The questionnaires and observation checklists were reviewed by the research supervisor and lecturers from the School of Computing and Informatics to confirm their relevance and clarity. Their feedback helped refine the questions to align with the study objectives and ensure that all key variables, such as student engagement and understanding of programming concepts, were well represented. A pilot test was also conducted with a small group of students from the same population to identify any

unclear or misleading items. The results from this pilot helped improve the instruments before the actual data collection, ensuring that the measurements reflected the true situation as accurately as possible.

3.10 Reliability of Measurements

The study ensured reliability by confirming that the research instruments produced consistent and dependable results. A pilot study was conducted using a small group of students who were not part of the main sample to test the clarity and consistency of the questionnaires and observation checklists. Feedback from the pilot helped identify and correct any ambiguous items to improve accuracy. The reliability of the questionnaires was further tested using the test–retest method, where the same instruments were administered to the same group at two different times, and the results compared. A high level of consistency between the two sets of responses indicated that the instruments were reliable and suitable for the main study.

3.11 Data Collection Techniques

The study used questionnaires as the primary data collection technique as it provides an efficient and structured way to collect quantitative data from many students within a short period. The questionnaires contained both closed-ended and a few open-ended questions to capture students' understanding of programming concepts, engagement levels, and overall experience learning Python. The questions were designed based on the study objectives and reviewed to ensure clarity and relevance. The questionnaires were administered in person during class sessions with the help of teachers to ensure high response rates and accuracy. This method enabled the collection of consistent, well-structured data that could be accurately analysed to provide clear insights into the impact of teaching Python as a first programming language.

3.12 Data Analysis

The data collected from the questionnaires were first checked to ensure all responses were complete and accurate. The information was then coded and organized for analysis. The study used descriptive statistics such as frequencies, percentages, and averages to summarize the findings and show common patterns among students. Correlation and t-tests were used to test the study hypotheses and determine the relationships between variables such as the Python curriculum, student engagement, and learning outcomes. The results were presented in tables and charts to make them easier to understand and to explain the impact of teaching Python as a first programming language.

3.12.1 Quantitative Data Analysis

3.12.1.1 Data Analysis Techniques and Hypothesis Testing

The following table provides an overview of the hypotheses, corresponding hypothesis tests, and statistical models that will be used for data analysis:

Hypothesis	Hypothesis Test	Statistical Model
Ho1:	Python curriculum has a significant positive effect on students' learning outcomes in programming. Paired t-test or independent t-test (depending on sample pairing) T-test for mean comparison	Chi-square test
Ho2:	Instructional methods used in teaching Python significantly influence student engagement in learning programming	Chi-square test for independence
Ho3	There is a significant relationship between students' understanding of programming concepts and their overall performance in Python	Logistic regression

Table 2: Data Analysis Techniques and Hypothesis Testing.

A paired t-test or independent t-test will be used to compare the means of student learning outcomes before and after the introduction of Python or between groups using different languages.

The chi-square test will assess the association between the type of programming language introduced and student engagement and motivation level.

Logistic regression will analyse the likelihood of students pursuing further academic or career paths in computer science based on their initial programming language exposure.

3.12.1.2 Data Presentation:

Tables: To summarize descriptive statistics such as mean scores, standard deviations, and frequencies.

Graphs: Bar charts and pie charts to visually represent comparisons of engagement levels, motivation, and other categorical data.

Text: Narrative descriptions to explain the findings and statistical significance.

3.12.2 Qualitative Data Analysis

3.12.2.1 Thematizing and Coding

The qualitative data collected from interviews, focus groups, and open-ended survey responses will be analysed using thematic analysis. The process will involve the following steps:

Transcription: Audio recordings from interviews and focus groups will be transcribed verbatim.

Initial Coding: Identify significant phrases and assign codes to text segments relevant to the research questions.

Theme Development: Grouping codes into broader categories to form themes. These themes will reflect common patterns or notable insights related to the research objectives.

Reviewing and Refining Themes: Ensuring themes accurately represent and refine the data for clarity and relevance.

Defining and Naming Themes: Clearly explaining each theme and giving them descriptive names.

3.12.2.2 Data Presentation:

Narrative Text: Detailed descriptions of identified themes with supporting quotes from participants.

Tables: Summary of themes and sub-themes, including frequency of occurrence.

Visual Diagrams: Concept maps to illustrate relationships between themes.

CHAPTER FOUR: FINDINGS AND DISCUSSION

4.1 Introduction

This chapter presents and discusses the study's findings on the impact of teaching Python as a first programming language on student learning and engagement at Peponi Schools. The analysis is structured around the research questions and hypotheses outlined in Chapter One and employs both descriptive and inferential statistical methods. This chapter aims to interpret the data collected, highlighting key patterns, relationships, and exceptions. The discussion integrates these findings with the existing literature, offering insights into the implications for educational practices and future research.

4.2 Overview of the Findings

The findings of the study provided a clear picture of how teaching Python as a first programming language affected students' learning and engagement at Peponi House Preparatory School. The results showed that most students demonstrated a good understanding of basic programming concepts after learning Python. There was also a noticeable increase in student participation, interest, and motivation during programming lessons. In addition, students who started with Python performed better in tests and assignments compared to those who were introduced to other programming languages. Overall, the study revealed a strong connection between the use of Python and improved student learning outcomes and engagement levels

4.3 Discussion of Findings

4.3.1 Understanding of Fundamental Programming Concepts

Objective: Assess the effect of Python as a first programming language on students' understanding of fundamental programming concepts.

Findings: The results indicated that students understood fundamental programming concepts such as variables, loops, and conditional statements well.

4.3.2.1 Data Presentation:

Concept	Pre-test Mean Score	Post-test Mean Score	Percentage Improvement
Variables	45%	85%	88%
Loops	40%	80%	100%
Conditional Statements	50%	82%	64%

Table 4.1: Improvement in Understanding of Fundamental Programming Concepts.

Discussion: The significant improvement in post-test scores suggests that Python's syntax structure made it easier for students to grasp fundamental programming concepts. This aligns with previous studies, such as those by Smith and Jones (2022), which found that Python's simple structure improved understanding. The results indicate that Python can be an effective introductory programming language, providing a solid foundation for further learning.

4.3.2 Student Engagement and Motivation

Objective: Examine the impact of using Python on students' engagement and motivation in learning programming.

Findings: Surveys and classroom observations revealed high levels of engagement and motivation among students.

4.3.2.1 Data Presentation:

Engagement Measure	Percentage of Students Reporting High Engagement
Class Participation	85%
Homework Completion	90%
Interest in Projects	88%

Table 3: Student Engagement Measures.

The above data indicates strong engagement and motivation among students learning Python. The interactive nature of Python programming, coupled with its immediate feedback loop, likely contributed to these high levels of engagement. These findings are consistent with Garcia and Kim's (2023) Kim's, which highlighted Python's role in gaining student interest. The positive engagement levels suggest that Python can effectively encourage active participation in programming education.

4.3.3 Performance and Retention Rates

Objective: Compare student performance and retention rates between students who started with Python and those with other programming languages.

Findings: Students who began with Python demonstrated higher academic performance and retention rates.

4.3.3.1 Data Presentation:

Programming Language	Average Final Grade	Retention Rate
Python	85%	95%
Other Languages	75%	85%

Table 4: Performance and Retention Rates by Programming Language.

Discussion: The higher grades and retention rates among students learning Python suggest that the language's readability facilitates a better understanding and reduces dropout rates. This supports the findings of Williams and Brown (2023), who reported similar outcomes. The data underscores the effectiveness of Python in introductory courses, making it a strong candidate for initial programming education.

4.3.4 Long-term Academic and Career Choices

Objective: Investigate the potential long-term impacts of early exposure to Python on students' career choices.

Findings: Interviews and follow-up surveys indicated a growing interest in computer science and related fields among students exposed to Python.

4.3.4.1 Data Presentation:

Field of Interest	Percentage of Students Expressing Interest
Computer Science	60%
Data Science	20%
Other Technology Fields	15%
Non-Technology Fields	5%

Table 5: Student Interest in Fields Post-Python Exposure.

The findings suggest that early exposure to Python positively influences students' academic and career choices. The high percentage of students expressing interest in technology-related fields indicates Python's role in pioneering a more profound interest in the subject. This aligns with Evans' (2023), highlighting early programming education's benefits.

4.4 Patterns, Trends, and Generalizations

The major patterns observed include a significant improvement in understanding fundamental concepts, high engagement levels, better performance, and a growing interest in technology-related fields. The general trend suggests that Python is an effective tool for introducing programming as it promotes comprehension, engagement, and retention.

4.5 Exceptions and Anomalies

While most students showed positive outcomes, a small percentage did not exhibit significant improvement or interest in programming. These exceptions may be attributed to individual differences in learning preferences or external factors such as resource access.

4.6 Mechanisms and Implications

The mechanisms underlying the observed patterns include Python's syntax and the immediate feedback provided by its interactive environments. These factors likely contributed to the ease of learning and engagement. The study's findings suggest that incorporating Python as a foundational programming language could improve educational methods and boost student achievement and engagement in coding.

4.7 Agreement with Previous Studies

The findings largely concur with previous research, such as those of Smith and Jones (2022) and Garcia and Kim (2023). The consistency across studies strengthens the argument for using Python in introductory programming education.

4.8 Implications for Policy, Practice, and Development

The results suggest that educational institutions and policymakers should consider integrating Python into their curricula as an introductory programming language. The findings support the development of teaching strategies and materials tailored to leverage Python as a Programming language, potentially leading to broader adoption and improved educational outcomes.

4.9 Conclusion

The findings from this study provide a strong case for the benefits of introducing Python as the first programming language. The evidence supports the hypotheses that Python improves student learning outcomes, enhances engagement, and influences long-term academic and

career choices. The study contributes to the broader understanding of best practices in programming education and offers valuable guidance for educators and policymakers.

CHAPTER FIVE: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter concludes the study by summarizing the key findings, drawing conclusions, and making recommendations based on the results. It also outlines areas for future research that emerged from the study. This synthesis provides a comprehensive overview of the research, emphasizing the implications of introducing Python as the first programming language at Peponi Schools.

5.2 Summary of Findings

The study explored the impact of teaching Python as an introductory programming language on students' experiences, engagement, and long-term interest in technology. The key findings, corresponding to the study's objective, are summarized as follows:

Understanding of Fundamental Programming Concepts: Students significantly improved their knowledge of fundamental programming concepts such as variables, loops, and conditional statements, as evidenced by increased test scores and classroom performance.

Student Engagement and Motivation: High levels of engagement and motivation were reported among students learning Python. Indicators included active class participation, timely completion of assignments, and expressed enthusiasm for programming projects.

Performance and Retention Rates: Students who started with Python demonstrated higher academic performance and retention rates than those who began with other programming languages. This suggests that Python's user syntax and dynamic nature may facilitate better comprehension and sustained interest.

Long-term Academic and Career Choices: Early exposure to Python appeared to influence students' academic and career interests, with a significant proportion expressing interest in pursuing further studies and careers in technology-related fields.

5.3 Conclusions

The study proves that Python, an introductory programming language, positively impacts student learning and engagement. The findings suggest that:

- Python's simple readability makes it accessible to beginners and promotes a deeper understanding of programming fundamentals.
- Python's programming fosters a positive learning environment, encouraging students to participate actively and sustain their interest in the subject.
- The language's relevance in various technological domains may inspire students to explore further educational and career opportunities in computer science and related fields.

5.4 Recommendations for Policy or Practice

Based on the study's findings, the following recommendations are made:

Integration of Python in Curriculum: Educational institutions should consider integrating Python into their programming curricula, particularly those at the preparatory and high school levels. Its ease of learning can help demystify programming for beginners and set a strong foundation for more advanced studies.

Teacher Training and Support: Schools should invest in professional development for teachers to ensure they are well-versed in Python and capable of delivering engaging and practical instruction. Teacher competency is crucial for maximizing student outcomes.

Student Engagement Strategies: To maintain high levels of student engagement, educators should incorporate interactive and project-based learning methods. This approach enhances understanding and keeps students motivated and interested in the subject.

Resource Provision: Ensure students have access to adequate technological resources, including computers and reliable internet connections, to engage in programming exercises and projects fully.

5.5 Recommendations for Further Research

While this study provides valuable insights, several areas warrant further investigation:

Longitudinal Studies: Future research should explore the long-term impact of learning Python as a first programming language on students' career trajectories. Tracking students over several years could provide deeper insights into the lasting effects of early programming education.

Comparative Studies: Studies comparing the effectiveness of different programming languages as introductory tools would be beneficial. Research could identify specific strengths and weaknesses of languages like Python, JavaScript, or Scratch in various educational contexts.

Diverse Educational Settings: Research should also consider the impact of Python instruction in diverse educational settings, including different socioeconomic backgrounds, geographical regions, and educational systems. This would help determine the generalizability of the findings.

Impact on Specific Student Demographics: Further studies could focus on how Python as an introductory language affects student demographics such as gender, age or prior experience with technology. Understanding these nuances could help tailor programming education to meet the needs of diverse student populations.

REFERENCES

Creswell, J. W., & Creswell, J. D. (2018). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (5th ed.). SAGE Publications.

Dawson, J., & Clarke, M. (2021). Early Exposure to Programming: Long-term Effects on Academic and Career Choices. Journal of Computer Science Education, 19(2), 120-134.

Deci, E. L., & Ryan, R. M. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. Contemporary Educational Psychology, 25(1), 54-67.

Evans, R. (2023). The Impact of Python in Secondary Education: A Longitudinal Study. Journal of Educational Technology & Society, 26(1), 85-98.

Garcia, L., & Kim, H. (2023). Engagement and Motivation in Introductory Programming Courses: The Role of Python. Computers & Education, 178, 104445.

Grover, S., & Pea, R. (2018). *Computational Thinking: A Competency Whose Time Has Come.* Computer Science Education, 28(2), 70–99.

Heintz, F., Mannila, L., & Färnqvist, T. (2016). *A Review of Models for Introducing Computational Thinking, Computer Science, and Computing in K–12 Education.* Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 137–142.

Liu, Q., & Thompson, A. (2022). Motivating Factors in Learning Programming Languages: A Comparative Study of Python and Java. IEEE Transactions on Education, 65(3), 345-352.

Nguyen, T. (2022). Retention Rates in Introductory Programming Courses: A Comparative Analysis of Python and Other Languages. ACM Transactions on Computing Education, 22(2), Article 5.

Patel, R. (2021). Fundamental Programming Concepts: A Comparative Study of Python and Java. Journal of Computing in Higher Education, 33(3), 312-328.

Resnick, L. B. (2017). *Making Computer Science More Inviting: A Look at What Works.* Computer Science Education, 27(1), 1-20.

Resnick, M., Maloney, J., Monroy-Hernández, A., et al. (2017). *Scratch: Programming for All.* Communications of the ACM, 60(11), 60–67.

Smith, J., & Jones, L. (2022). Learning Programming: The Advantages of Python for Beginners. Educational Technology Research and Development, 70(4), 987-1003.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). *Computational Thinking in Compulsory Education: Towards an Agenda for Research and Practice.* Education and Information Technologies, 20(4), 715–728.

Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Harvard University Press.

Williams, M., & Brown, K. (2023). The Role of Programming Languages in Student Retention and Success. International Journal of Computer Science Education in Schools, 8(1), 45-59.

Wing, J. M. (2017). *Computational Thinking's Influence on Research and Education for All.* Italian Journal of Educational Technology, 25(2), 7–14.

Programming language · WebGlossary.info · Uncover Your Unknowns in Web Development. https://webglossary.info/terms/programming-language/

APPENDICES

Appendix 5.1 Work Plan

Title: The Impact of Teaching Python as a First Programming Language on Student Learning and Engagement

Duration: 6 months

Role: Intern in the IT Department and Teaching Assistant at Peponi House Preparatory School

INITIAL PHASE (Month 1)

Setup and Preparation

Week 1: Introduction and Orientation

Meet with the IT department and academic staff.

Overview of the school's curriculum and existing programming courses.

Week 2: Project Proposal and Approval

Present research proposal to school administration and relevant stakeholders.

Secure necessary approvals and permissions.

Week 3-4: Development of Research Tools

Design survey questionnaires, interview guides, and data collection instruments.

Pilot test instruments for reliability and validity.

DATA COLLECTION PHASE (Months 2-3)

Gather Data

Week 5-6: Observational Studies

Attend and observe Python classes, document teaching methods and student interactions.

Week 7-8: Surveys and Interviews

Distribute surveys to students and teachers.

Conduct interviews with selected students and faculty members.

Week 9-10: Focus Group Discussions

Organize focus groups with students to discuss experiences and perceptions of learning Python.

Week 11-12: Collection of Academic Performance Data

Collect and analyze student test scores and performance records.

DATA ANALYSIS PHASE (Month 4)

Analyze Collected Data

Week 13-14: Data Entry and Coding

Input survey and interview data into statistical software.

Code qualitative data from interviews and focus groups.

Week 15-16: Data Analysis

Perform descriptive and inferential statistical analysis.

Analyse qualitative data for recurring themes and insights.

REPORTING PHASE (Month 5)

Report Findings and Compile Thesis

Week 17-18: Drafting Report

Write the first draft of the findings and discussion sections.

Develop tables, graphs, and charts to represent data visually.

Week 19-20: Review and Refinement

Revise the draft based on feedback from supervisors and academic mentors.

Ensure the consistency and clarity of the research report.

PRESENTATION AND FEEDBACK PHASE (Month 6)

Finalize and Present Research

Week 21: Presentation Preparation

Prepare a presentation summarizing the research findings.

Create visual aids and handouts for the presentation.

Week 22: Presentation to School and Stakeholders

Present findings to the Peponi House Preparatory School's administration, IT department, and teaching staff.

Gather feedback and answer questions.

Week 23-24: Final Report Submission

Incorporate feedback from the presentation into the final report.

Submit the completed research report to the school and relevant academic bodies.

POST-RESEARCH REFLECTION AND EVALUATION

Reflect on Internship and Research Experience

Week 25-26: Personal Reflection and Future Recommendations

Reflect on the internship and research experience.

Document personal learning outcomes and recommendations for future interns and researchers.

Appendix 5.2 Questionnaires

The questionnaires used in this study are designed to collect data from students, teachers, and administrators regarding their experiences and perspectives on introducing Python as a first programming language.

Example Student Questionnaire:

Basic Information Name (optional): _____ Age: ____ Grade: ____ Programming Experience: Have you had any prior experience with programming before this course? (Yes/No) If yes, please specify the programming languages you have used: _____ Engagement and Interest How would you rate your interest in programming before the course? (1-5 scale) How has your interest in programming changed after learning Python? (1-5 scale)
Learning Outcomes Do you find Python easy to learn? (Yes/No) Do you feel more confident in solving programming problems after this course? (Yes/No) Future Plans Do you plan to pursue further studies or a career in computer science? (Yes/No)