

COMPARING C++, JAVA, AND PYTHON

by

Jayla Raine Wyandon

Research Paper Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Computer Information Systems

Northwestern State University, School of Business

May 2025

COMPARING C++, JAVA, AND PYTHON

by

Jayla Raine Wyandon

A Thesis Proposal

Submitted to the Graduate School of

Northwestern State University of Louisiana

In partial fulfillment of requirements for

Master's degree in computer information systems

Approved by:

Dr. Curtis Penrod

Dr. Begona Perez-Mira

Dr. Sarah Wright

Greg Handel, D.M.A.
Dean, The Graduate School

Abstract

Choosing a programming language for a task, project, or learning how to program can be difficult. Each programming language uses different variables, functions, writing styles, and methods. This research will focus on three programming languages, C++, Python, and Java, with the purpose of comparing readability, speed, syntax, and portability to determine the advantages and disadvantages of each language. Each language will run the Fibonacci sequence to compare these factors and allow for an easy way to view the differences in these languages. This research shows that each language has its own advantages and disadvantages and it is important to be able to understand these factors to determine which language will work best for a user or task.

Keywords: C++, Java, Python, programming, comparison

Table of Contents

COMPARING C++, JAVA, AND PYTHON.....	ii
Abstract.....	iii
List of Tables	vi
List of Figures.....	vii
Chapter 1: Introduction.....	1
Introduction to the problem	1
Statement of the problem.....	1
Purpose of the study	2
Research questions	2
Significance of the study.....	3
Definition of Terms.....	3
Assumptions, Limitations, and Delimitations.....	4
Conclusion.....	4
Chapter Two: Literature Review	5
Introduction.....	5
Factors Used in Comparison of Programming Languages	5
Most Popular Programming Languages.....	9
Programming languages chosen for the study.....	12
Conclusion.....	14

Chapter Three: Methodology	15
Introduction.....	15
Research Design.....	15
Research Questions and Hypothesis	16
Data Collection.....	16
Data Analysis.....	18
Conclusion	18
Chapter 4: Results.....	19
Results.....	19
Summary	24
Chapter 5: Conclusion	25
Study Results	25
Recommendations	26
Summary	27
References	28

List of Tables

Table 1: Total number of words borrowed from the English language.....	21
Table 2: Lines of code used to write the Fibonacci sequence.....	21
Table 3: Total time to calculate 20 numbers.....	23
Table 4: Total time to calculate 200 numbers.....	23

List of Figures

Figure 1: Total time to Calculate 20 Numbers.....	23
Figure 2: Total Time to Calculate 200 Numbers.....	24

Chapter 1: Introduction

Introduction to the problem

Learning a new programming language can be time-consuming especially for those who have never learned a new programming language (Saito & Yamaura, 2013). Some authors determined that the chosen programming language and how a programmer learns the concepts can lead to difficulties in learning a new language (Cazzola & Olivares, 2016). Each programming language consists of many different features. Examples of programming features include variable definition, functions, and methods. Examples of programming languages include C++ and Java, which are popular programming languages in both business and education (Cazzola & Olivares, 2016). According to a previous study reviewing the most popular programming languages, the most common programming and scripting languages used are JavaScript, Python, Java, C++, PHP, and C# (Đurđev, 2024). Efficiency is a goal of many programs. Efficiency is important when working on any project, because 60% of a project is spent on data cleansing and organizing, and inefficient processing of data can lead to time being wasted (Beierlieb et al., 2023).

Statement of the problem

The problem addressed in this study is the difficulty deciding which programming language to choose when starting a project. Programmers have several different programming, scripting, and markup languages from which to choose when working on a project. Examples of scripting and markup languages include Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (Park & Wiedenbeck, 2011). Examples of programming languages include Python, Java, and C++ (Ali & Qayyum, 2021). Because there are many

programming languages, programmers should understand which language would work the best for a given task.

Purpose of the study

The goal of this replication study is to compare the differences in three programming languages by replicating the studies *Performance Comparison of Most Common High Level Programming Languages* (Parveen & Fatima, 2016) and *A Pragmatic Comparison of Four Different Programming Languages* (Ali & Qayyum, 2021). The study by Parveen and Fatima compared C, C++, and Java. The study by Ali and Qayyum compared C, C++, Python, and Java. Languages in this study will be C#, Java, and Python. The results of this study may allow readers to learn about the difference in programming languages to determine what languages beginners can use when learning how to design applications and complete tasks. By comparing the results of a task completed with different languages, it will be easier to view the advantages and disadvantages of each language.

Research questions

The research questions for this study are as follows:

- What are the differences in syntax between programming languages in this study?
- What are the differences in processing time and code size between programming languages in this study?
- Which programming language used in this study uses the least amount of code as well as having the shortest run time?

Significance of the study

The significance of the study is to be able to identify the differences in three popular programming languages as well as determine the strengths and weaknesses that come with each language. Analyzing the differences between languages will also help students learn the advantages and disadvantages that different languages have. This research will also help students in understanding the characteristics of these languages and help them to choose what language to use for a particular project.

Definition of Terms

Key definitions for this study include:

- **C++:** C++ is a machine independent object-oriented language that is able to run on most operating systems. C++ was developed due to the language C not having any object-oriented features and has now existed for around 30 years and has many online resources available (Aljas et al. 2023) C++ can be used to create applications such as mobile apps and websites.
- **Java:** Java is an object-oriented programming language that is used to develop mobile apps, web apps, desktop apps, and games. Java is platform-independent, which allows it to run on any software as well as have an accessible way to extract and share data (Martinez et al., 2023)
- **Python:** Python is an open-source object- oriented programming language that is used for general-purpose software engineering (Huang, 2022). Python is known as one of the easiest programming languages to learn and its main features include being free to use, platform independent, both procedure oriented and object oriented, and having an extensive library (Cutting & Stephen, 2021).

Assumptions, Limitations, and Delimitations

Assumptions are a set of beliefs and values from prior experiences in a field of study (Creswell, 2018). An assumption for this study is that programming languages with the most learning resources available will be more popular than the languages that have less resources available. Another assumption would be that older programming languages such as C++ would use a lot of code to accomplish one task. Another assumption could be each programming language will have different methods to complete the same task.

Limitations are weaknesses in a research study that are often attached to the methods of a study (Creswell, 2018). Possible limitations for this study include the choice and number of comparisons of the programming language. Delimitations are limitations that specify the scope of a study and help further define the parameters of a research study (Creswell, 2018). A delimitation in this study is that there will only be three languages tested in this study: C++, Java, and Python.

Conclusion

The problem addressed in this study is the difficulty deciding which programming language to choose when starting a project. Learning a new programming language can be difficult and the goal of this replication study is to compare the differences in three programming languages that can be used in application development. Because there are a lot of different programming languages, the focus of this study will be on languages used in application development, which are C++, Java, and Python.

Chapter Two: Literature Review

Introduction

The purpose of this study is determining the differences in programming languages. The reason for this choice of topic was to find methods that can be used to compare programming languages. Many different programming languages exist, and analyzing the differences in each programming language can help determine the advantages and disadvantages of each language. This literature review includes different scholarly sources that were used to determine what factors will be replicated in this study.

This literature review also includes previous studies that compared different programming languages. Other literature that will be included in this review focuses on defining and determining the most popular programming languages. An important note is that some researchers included scripting and other languages in their studies despite differences in scripting and programming languages.

Factors Used in Comparison of Programming Languages

There are different methods that are used to compare different programming languages. Previous scholarly articles begin by listing information about each language used in the study and comparing different features. In the study *Performance Comparison of Most Common High Level Programming Languages* by Zahida Parveen and Nazish Fatima (2016), the three programming languages chosen were Java, C#, and C++. The authors created a table so readers would be able to understand the differences and similarities each language had. For example, for object orientation, all three languages are hybrid with the exception of C++, which is also multi-paradigm. All three languages are static typing, and all three use classes, variables, and methods.

Programming languages can be compared through analyzing the different ways code is written. In the article *Performance Comparison of Most Common High Level Programming Languages* (2016), Zahida Parveen and Nazish Fatima created a program that outputs the Fibonacci Series for each language. This method showed the different ways of writing code in each programming language while being able to accomplish the same goals. For example, to display results Java requires the command “System.out.print,” C# uses the command “Console.WriteLine” and C++ uses the “cout” command. The study *A Pragmatic Comparison of Four Different Programming Languages* (2021) by Saqib Ali and Sammar Qayyum also compares the different ways code is written across different languages, but instead compares the programming languages C, C++, Python, and Java. In this study, the authors utilized each language to create the message “Hello World” in order to compare syntax. The authors were able to determine that Python required the least amount of code when displaying one sentence while Java required the most code.

Parveen and Fatima (2016) also compared the length of code required to accomplish the same task for different languages. While Ali and Qayyum (2021) manually counted the lines of code used in each compiler, Parveen and Fatima (2016) used a software metric to determine the size of a computed program. Parveen and Fatima (2016) also used their methods to compare the syntax of each language with natural language. This comparison allowed for the authors to determine which language had the best readability. The study *Comparison of python and java for use in instruction in first course in computer programming* by Ogbuokiri et al. (2016) also compared the length of code required by Java and Python to complete a task through creating the same algorithm and comparing the code size. Alomari et al. (2015) also compared the different

ways each language is written but does not compare the different methods each language has when completing the same task.

Ogbuokiri et al. (2016) compared the difference in commenting between Python and Java, which can help determine which program is the most efficient in readability and commenting. The authors state that Java encourages comments to be written in multiple lines by separating comments with asterisks, while Python only needs one line with the use of one #.

Several authors compared the efficiency and speed of multiple languages. Ali and Qayyum (2012) determined that efficiency and speed of each language is determined by the dynamic memory access, array item iteration, and integer and float calculations. The authors calculated the average running time and CPU usage of each language five times. Parveen and Fatima (2016) determined the speed and efficiency through execution time and compilation time. Ogbuokiri et al. (2016) compared running and execution times by running the same program in Python and Java and timing it to determine the running time.

Parveen and Fatima (2016) compared memory management separately from speed and analyzed each language's methods of handling memory loss and how frequently memory leaks occur in each language. Alomari et al. (2015) also compared memory management by explaining how each language handles allocation of memory. For example, the authors stated how C# uses one method to automatically manage memory resources when running in comparison to C++ having different techniques to manage memory. Ogbuokiri et al. (2016) compared memory consumption, and used a method where Python and Java are installed on the same machine in order to determine how much memory is consumed by both.

Another method of comparing different programming languages includes listing the advantages and disadvantages of each language studied. Alomari et al. (2015) compared six different programming languages: C++, PHP, C#, Java, Python, and VB with the goal of determining the advantages and disadvantages of each. For example, while C++ is easy to learn and use, it takes up more memory space than other languages. In this study the chosen languages were compared under the characteristics of reusability, reliability, portability, availability of tools, readability, efficiency, familiarity, and expressiveness. When comparing readability and expressiveness, the authors discussed the different ways to write code for each program. Efficiency was determined by the memory available when using each program. This study did not use one method to compare each language and instead discussed the difference in syntax, memory management, available IDEs, compilation, and execution.

Another study that lists the advantages and disadvantages of programming languages is *Comparison of Programming Languages: Review* by Reddy et al. (2018). Similar to the study *Comparative Studies of Six Programming Languages* (2015), when comparing different languages, the authors listed the advantages and disadvantages of each. Some of the advantages and disadvantages relates to memory consumption, user interface, and readability.

Another method that can be used to compare programming languages is by determining which programming language would be the best for processing a large amount of data. A previous study that uses a different method to compare programming languages is *Choosing the Right Programming Language in Making a Website Backend Using the Waterfall Method* by Tegh Rijanandi and Faisal Dharma Adhinata (2022). This study is different to the previously mentioned studies because it uses the waterfall method to compare different programming languages. This method consists of five steps: requirements, design, implementation, testing, and

maintenance. The requirements of the website was to be able to process a large amount of user information. The information included name, ID number, gender, and address for each resident. The design stage consisted of building a website that included a form for users to input the required information. The implementation stage consisted of performing performance tests on the four languages chosen (PHP, Python, Java, and C#) with the purpose of determining which programming language would be the best when creating a website backend to process data. After each language was implemented for the backend of the website, the testing stage allowed for the authors to gather data on which language has the best performance when processing data. The authors collected 2423 population data that each program had to find, manage, and process.

There are different methods that are used to compare programming languages. The categories that were previously used to compare programming languages include readability, efficiency, speed, and memory management. Other methods that are not commonly used include comparing how languages handle errors and the different ways users can write comments. Previous studies also compared similar programming languages. For example, Java is compared to C++ in *Comparative Studies of Six Programming Languages* (Alomari et al., 2015) and in *A Pragmatic Comparison of Four Different Programming Languages* (Ali and Qayyum, 2024). When deciding on the languages that will be compared, past literature is reviewed to determine the most commonly used languages.

Most Popular Programming Languages

This study will focus on programming languages that are commonly used in a technological environment. There are several ways to determine the most popular programming languages. One way is by creating a survey and analyzing the data gathered. Dada et al. (2021) determined the most used programming languages used in information technology by analyzing

data from multiple surveys. The authors stated that the community StackOverflow is one of the most active communities for developers to share their knowledge, so data was gathered from the yearly surveys that StackOverflow creates. Around 65,111 users took the survey in 2019. The authors stated that in 2019, the most popular programming and scripting languages included JavaScript, HTML/CSS, SQL, Python, Java, PowerShell, C#, and PHP.

A Comparative Analysis of Programming Language Preferences Among Computer Science and Non-Computer Science Students by Islam et al. (2024) is another study that uses a survey to gather data about the most commonly used programming language, but instead of gathering data from an online survey, the data was gathered through 500 students at a university with the goal of creating a comparative analysis of programming language preferences among computer science and non-computer science students. The survey separated information through continent of origin, academic major, and programming language preference. The study concluded that for the university, Python was the most popular language followed by JavaScript, C, other languages, C++, and Java. The study also showed that while some languages, such as C#, were commonly used for applications, they might not be preferred by users.

Another way of determining the popularity of programming languages is by analyzing different trends. The article *Learning a Programming Language for Today* by Tabusca Alexandru (2015) is a study analyzing several different trends and issues of the field of Internet Technology as well as programming languages. The method Alexandru used when analyzing trends was comparing data gathered from TIOBE and GitHub. The TIOBE index is an indicator of the popularity of programming languages, and the ranking is updated every month based on the number of experienced programmers, business applications, and third-party vendors on the internet (Zhang et al., 2020). According to TIOBE index, the top five languages are Java, C,

C++, C#, and JavaScript. GitHub is one the largest publicly available code repository with almost three and a half million users (Alexandru, 2015). The top five languages for GitHub include JavaScript, Java, Python, CSS, and PHP.

The literature *Popularity of Programming Languages* by Darko Đurđev (2024) is another study that focuses on ranking the popularity and relevance of programming languages in 2024. While the previously discussed studies only gathered data from one website, Đurđev gathered data through surveys from multiple websites with the goal of analyzing programming trends and their real-world usage by developers. Đurđev first determined how often language tutorials are searched on Google, and states that the most relevant languages are Python, Java, JavaScript, C++, C#, R, and PHP. Based on the TIOBE index that is used to rank online searches across multiple engines, the most popular languages are Python, C++, Java, C, C#, and JavaScript. Lastly by analyzing Stack Overflow surveys, the most popular programming languages are JavaScript, SQL, HTML/CSS, Python, TypeScript, Java, C#, C++, and PHP. Similar to the study by Alexandru, this study used the TIOBE index to help determine the currently most used programming languages.

Analyzing the information gathered from studies that analyzed the most used programming languages to studies that compared different programming languages to find advantages and disadvantages of each, there are several studies where authors chose to compare programming languages that are commonly used in information systems. The study by Alomari et al. (2015) mentioned that while there are thousands of different programming languages but only four popular programming languages will be compared. These languages are not only popular but have a wide range of use. The studies by Qayyum and Ali (2021), Parveen and Fatima (2016), and Ogbuokiri and Agu (2016) also chose popular languages to compare. The

study by Rijanandi and Adhinata (2022) also chose popular programming languages when creating a backend website. This study will focus on comparing commonly used programming languages as well.

This study will replicate the previous studies *A Pragmatic Comparison of Four Different Programming Languages* (2021) and *Performance Comparison of Most Common High Level Programming Languages* (2016). The languages chosen for these studies are C, C#, C++, Java, and Python. This study will compare Python, Java, and C++.

Programming languages chosen for the study

After choosing the three languages used in this study, a determination was made regarding whether each programming language could be compared to the other languages.

Python is a dynamic typed language but can also be used as a scripting language used in developing standalone applications such as websites (Cutting & Stephen, 2021). Python is also free to use, open source, platform-independent, portable, and both procedure-oriented and object-oriented. Python is known as being easy to learn because the syntax is straightforward (Cutting & Stephen, 2021). Python is portable because it is compatible with a wide variety of hardware such as Linux, MacOS, and Microsoft Windows (Cutting & Stephen, 2021).

There are disadvantages to Python. When compared to Java, C, and C++, it runs slower. Python also consumes a large amount of memory when running, which makes it less favorable to use on mobile devices (Cutting & Stephen, 2021).

The document *Python for Web Development* by Dr. Uday and Uday Parkar (2022) is a study on the language Python. While Cutting and Stephen (2021) discuss the advantages and disadvantages of Python, Uday and Parkar (2022) explain how Python is a high-level general-

purpose programming language. Python's language allows programmers to write clear and logical code for any type of project, including web development and game development (Uday & Parkar, 2022).

Java is a class-based, object-oriented, and high-level programming language (Martinez et al., 2023). Java is known for being simple to use, portable, and platform-independent, and these characteristics led to Java being a popular choice in both industries and schools when developing applications such as mobile apps, web apps, desktop apps, and games. The authors Martinez et al. (2023) state several different features of Java in their study, *A Review on Java Programming Language*. Similar to Python, Java is object-oriented, which means all items are displayed as objects. An object is an instance of a group that can contains attributes and functions. This allows for data to be grouped into categories. Java is also simple to use because developers are able to write code and run it on any platform that supports Java (Martinez et al., 2023). Java's platform independence comes from being able to provide the resources needed to run on a software and having an accessible way to extract and share data (Martinez et al., 2023).

C++ is an object-oriented programming language that was developed after the language C because C did not have object-oriented features (Aljas et al., 2023). Another reason C++ is chosen over C is due to its feature of being able to manage memory spaces efficiently (Singh et al., 2021). Similar to Java and Python, C++ is an object-oriented program used for website and game development. C++ is a popular programming language for a number of reasons. Because C++ is a high-performance language, it has effective memory management to help run programs quickly. Because C++ has been around for over thirty years, it consists of a large community of developers and online resources (Aljas et al., 2023). While C++ is not platform-independent, it is machine-independent, which allows C++ to run on most operating systems.

Like Java and Python, C++ is case-sensitive, which means that each word can be written multiple times and still be distinct. Characteristics that C++ has includes object-oriented programming, portability, and flexibility. These features are also included in Java and Python, which is why all three languages are popular choices when programming (Foster, 2014).

Previous literature not only chose languages that were commonly used and popular, but also shared similar features and are able to complete the same tasks. There are several features that Java, Python, and C++ share. All three programming languages are object-oriented, free to use, platform-independent, portable, and dynamically-typed. While each language is written differently, several authors state that all three programs are easy to use and include syntax that is easy to read.

Conclusion

The goal of this study is to compare programming languages. There are several different methods that were used in previous studies to compare languages. Examples in previous studies include comparing the memory, execution time of a program, and lines of code. By reviewing different methods used in previous studies, the studies *Performance Comparison of Most Common High Level Programming Languages* and *A Pragmatic Comparison of Four Different Programming Languages* were chosen to replicate. After deciding on a study to replicate, three different programming languages were chosen after reviewing the most commonly used programming language. The most common languages used in technology included Python, Java, JavaScript, C++, C#, R, and PHP, and this study will compare C++, Java, and Python. Previous studies were also reviewed in order to gather more information about each programming language.

Chapter Three: Methodology

Introduction

In the past, previous researchers compared two different programming languages by comparing the memory consumption, program length and execution time to complete the same task, and how each program handles errors (Ogbuokiri, 2016). Another study included the comparison of several programming languages by listing their strengths and weaknesses, the process used for executing code, typing strategies, and available IDEs (Alomari et al., 2015).

When analyzing data in the past, previous researchers presented code excerpts and commented on the differences in the case of teaching and learning that were observed (Pellet et al., 2019). One example compared code used in both Java and Python to print a sentence. The purpose was to show the concepts of storing a value in a variable and how to print a message to the console. This example showed that the phrasing would be different even though the goal is the same (Pellet, 2019).

Research Design

This study will replicate the research methodologies *Performance Comparison of Most Common High Level Programming Languages* by Zhida Parveen and Nazish Fatima (2016) and *A Pragmatic Comparison of Four Different Programming Languages* by Saqib Ali and Sammar Qayyum (2021). Parveen and Fatima (2016) used the Fibonacci sequence to compare three different programming languages: C++, C#, and Java. Ali and Qayyum compared C, C++, Python, and Java (Ali, 2021). Along with C++, C#, and Java, Python is compared as well since the language became one of the most widely used languages in the field of Artificial Intelligence and Data Science (Ali, 2021). For this study, three programming languages will be compared: C+, Java, and Python.

Research Questions and Hypothesis

The research questions for this study are as follows:

- What are the differences in syntax between programming languages used in this study?
- What are the differences in processing time and code size between programming languages in this study?
- Which programming language used in this study uses the least amount of code as well as having the shortest run time?

This study will attempt to discover if there is a relationship between the code size and execution time for each programming language, and determine if each programming language has different advantages and disadvantages

Data Collection

In both studies by Zhida Parveen and Nazish Fatima (2016) and Aqib Ali and Sammar Qayyum (2021), a common program was chosen to compare three languages. The criteria that will be used to compare each language are syntax, lines of code, machine dependency, and execution time.

The program “printing Fibonacci Series” will be executed in each programming language. Fibonacci numbers are a series of numbers where each number in the sequence is given by the sum of the previous two numbers. The Fibonacci series is also used in applications such as computer algorithms and financial models (Liu, 2023). Parveen and Fatima used the Fibonacci sequence to compare three different programming languages. Using the same method for each language allows for a clear comparison of syntax. Lines of code (LOC) is a software metric that is used to determine the size of a computed program (Parveen & Fatima, 2016). The

size of the program can be estimated by counting the number of lines the code has while written as text. Lines of Code (LOC) is used to determine the size of code in a program by counting each physical source line of code in a program, excluding blank lines and comments since comments and empty lines do not affect how a program runs (Zhang, 2009).

To compare the readability of the syntax of each programming language, Parveen and Fatima (2016) compared how many words in each programming language use terms taken from the English language. The same comparison will be used in this study.

Machine dependency is determined by the portability of a programming language and if a language relies on a platform to compile code (Parveen & Fatima, 2016). Java is machine independent because Java has its own compiled code that can run on all operating systems. When using Java, the program will be first compiled by Java Virtual Machine into Java byte code. Then the bytecode will be converted into the user's machine language (Elvis, 2014). C++ is machine independent. While C++ does not have its own code that is interpreted, it is available on all major operating systems (Elvis, 2014). Python can run on a wide variety of platforms and has the same interface on all platforms (Elvis, 2014).

Execution time is the amount of time taken to execute a file to produce output. To compare execution time, Zhida Parveen and Nazish Fatima (2016) used System namespace diagnostics for C#, and the method `getTime()` for Java and C++. Speed and efficiency are determined by the execution time. For the study by Ali and Quayyum (2021), execution time was compared for both one short and a large calculation five times to determine the average. This study will be using the method by Ali and Quayyum (2021).

Data Analysis

To analyze each programming language, for loops will be utilized, and each program will output the Fibonacci series up to the 20th number. In programming, a for loop is a method that repeats a set of code until a requirement is met. A for loop contains three statements. The first statement creates a number, the second statement is the requirement that needs to be met, and the third statement is executed after the written block of code is executed. An example in Java would be:

```
for (int n = 0; n < 3; n++) {  
    System.out.println(n);  
}
```

Where $n = 0$ is the first statement, $n < 3$ is the second statement, and $++n$ is the third statement. The reason why these methods are chosen is to replicate the study *Performance Comparison of Most Common High Level Programming Languages* by Zhida Parveen and Nazish Fatima (2016), where for loops are used to compare three languages. This method will be used to determine the run time for each language, and this method will also be used to determine the Fibonacci sequence up to the 20th number and 200th number to compare times for a small and large calculation.

Conclusion

The hypothesis of this paper states that there is a relationship between the code size and execution time for each programming language, and each programming language has different advantages and disadvantages. The three chosen languages for this study include C++, Java, and Python. This study will replicate the design by Zhida Parveen and Nazish Fatima (2016), which uses code to produce the Fibonacci method to compare three programming languages. The

factors that will be compared include syntax, lines of code, machine dependency, and execution time.

Chapter 4: Results

Results

For this study a method was created that consisted of one for loop to output the Fibonacci sequence for each language using as minimal code as possible for each language.

The code for C++ is as follows:

```
#include <iostream>
using namespace std;
int main() {
    int a = 0;
    int b = 1;
    for (int i = 0; i < 20; i++) {
        cout << a << "\n";
        int c = a + b;
        a = b;
        b = c;
    }
    return 0;
}
```

The code for Java is as follows:

```
class Main {
    public static void main(String[] args) {
        int n = 20, a = 0, b = 1;
```

```
for (int i = 1; i <= n; ++i) {  
    System.out.println(a);  
    int c = a + b;  
    a = b;  
    b = c;  
}  
}  
}
```

The code for Python is as follows:

```
a, b = 0, 1  
for x in range(20):  
    print(a)  
    a, b = b, a + b
```

In this study, Python contained the several terms that are similar to English. By using simple commands such as “in range” and “print”, the language allows for readers to understand the meaning of the given program. C++ also used several terms that are similar to English, but also used several other terms that might require readers to learn the meanings of the given commands. An example might be cout, which stands for character output. While Java uses simple terms such as int for integer, but also uses terms that beginners might not know the meaning of for example ‘public static void main(String[] args).’ The chart below compares the number of terms used in each language for this study.

Table 1: Total number of words borrowed from the English language

Number of Words Borrowed from the English Language		
C++	Java	Python
6	6	4

The total lines of code for each language are compared to determine which language is the most efficient when programming. For this study, Python uses the least amount of code to complete the given task. Even when counting the number of words instead of lines for C++ and Java, Python will still be the most efficient writing language. Figure 1 shows the lines of code used to complete the task in each language.

Table 2: Lines of code used to write the Fibonacci sequence

Lines of Code		
C++	Java	Python
13	12	4

When comparing machine dependency for each programming language, Java has the best portability due to being able to compile code into its own language in order to run on any platform. Java has three types of portability: source code portability, CPU architecture portability, and operating system portability (Foster, 2014). C++ is portable due to being available on major operating systems, but because not all of C++ compilers use the same standard, the portability may change (Foster, 2014). Python has better portability than C++ due to having many third-party libraries that work together to increase its portability (Foster, 2014).

To determine the execution time for each program, the method `std::chrono::system_clock::now();` is used for C++, `System.nanoTime()` is used for Java, and `datetime.now()` is used for Python. Each statement is stated at the beginning and end of each sequence to determine the amount of time each program uses. Replicating the study by Ali et al.

(2021), the average of 5 times is compared. For this study, the device used was an HP laptop with Windows 10 installed, AMD Ryzen 3 64-bit processor, and 12.0 GB ram. 45% of memory was being used during each test.

C++:

```
#include <iostream>
#include <chrono>
#include <ctime>

auto start = std::chrono::system_clock::now();
    // calculations
auto end = std::chrono::system_clock::now();
std::chrono::duration<double> elapsed_seconds = end-start;
std::time_t end_time = std::chrono::system_clock::to_time_t(end);
std::cout << elapsed_seconds.count()
    << std::endl;
```

Java:

```
long start = System.nanoTime();
**calculations**
long finish = System.nanoTime();
long timeElapsed = finish - start;
System.out.println(timeElapsed);
```

Python:

```
import datetime
dt = datetime.datetime.now()
print(dt)
```

Table 2 and figure 1 show the running time to each language took to calculate 20 numbers in the Fibonacci sequence. Table 3 and figure 2 show the running time used to calculate 200 numbers.

Table 3: Total time to calculate 20 numbers

Running Time in Seconds for 20 Numbers			
Language	C++	Java	Python
Iteration 1	0.00007574	0.00090728	0.000101
Iteration 2	0.00007319	0.00112161	0.000238
Iteration 3	0.000086489	0.0010227	0.000166
Iteration 4	0.00008746	0.00082438	0.000136
Iteration 5	0.00006722	0.00091723	0.000172
Average	0.0000780198	0.00095864	0.0001626

Figure 1: Total time to Calculate 20 Numbers

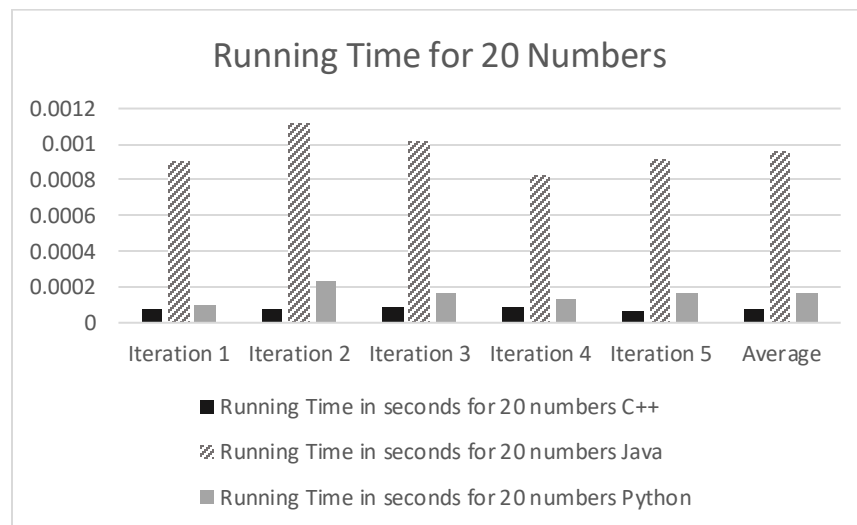
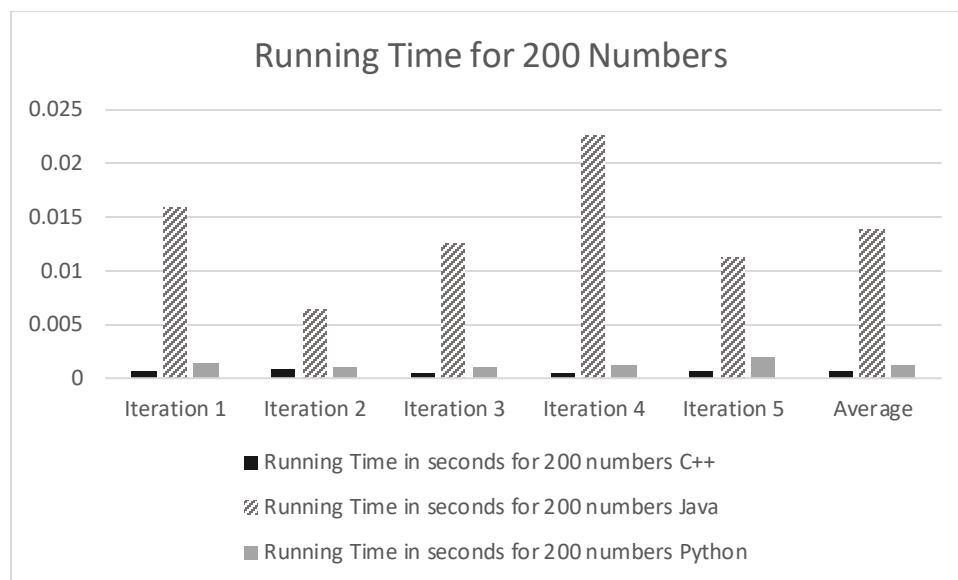


Table 4: Total time to calculate 200 numbers

Running Time in Seconds for 200 Numbers			
Language	C++	Java	Python
Iteration 1	0.00070063	0.015972	0.001334
Iteration 2	0.000802879	0.0065271	0.000985
Iteration 3	0.00048555	0.012636	0.001075
Iteration 4	0.00055232	0.022661	0.001307
Iteration 5	0.00059911	0.011391	0.001898
Average	0.000628098	0.01383742	0.0013198

Figure 2: Total Time to Calculate 200 Numbers



The results from these graphs show that C++ has the shortest execution time followed by Python and Java. The execution time for Python is very similar to C++ and is only slower by a small amount. When calculating a larger amount of numbers all languages require more time but the order of fastest to slowest language remains the same.

Summary

When completing a task, Python uses the least number of lines of code as well as being the simplest language in comparison to C++ and Java. Java is the most machine-independent because of its portability. Java also has the longest running time for both small and large calculations in comparison to C++ and Python. C++ has the shortest running time for calculations. This study shows that Java is best for portability, C++ would be best for speed, and Python would be efficient due to short run times as well as using the least number of lines of code for a given task.

Chapter 5: Conclusion

Study Results

This study allows readers to see the different advantages and disadvantages of each language. The results of this study may allow readers to learn about the difference in programming languages to determine what languages beginners can use when learning how to design websites or other applications. This study also allows readers to decide which factors of a programming language will work best in a project.

In terms of syntax for this study, the Python language is written similarly to the English language. Commands written in Java also use words taken from the English language like Python but also consists of statements and commands that might take a while to understand the meanings of. The C++ language uses terms that Java also uses, but C++ and Java both use different methods to display answers. While this study can be used to compare readability, readability also depends on the user. One user might find Java easier to understand while another user might understand Python more.

While C++ used the most amount of code in comparison to Java and Python, C++ took the least amount of time to run the equation, followed by Python then Java with the most time taken. This shows that the amount of code used in a program does not always lead to a shorter run time. When comparing both run time and code size, Python would be the best option due to using the least amount of code when writing the equation and taking a short amount of time to run the statement as well.

This study has the goal of determining if there is a relationship between the length of code written and the time it takes to execute. While this study focuses on the variables of syntax and amount of code used, other variables in this study that could have an effect on the execution

time of a program includes a device's central processing power (CPU) and the programmed task used for comparison.

Strengths of this study include learning about methods to compare and analyze programming languages. Limitations of this study include only analyzing three languages when there are other languages available that could be better options to use depending on a situation, and only comparing four different factors of a language when other factors could be analyzed as well.

Recommendations

This study could be replicated with other programming languages because this study only uses three when there are many languages to choose from. While these languages are popular choices in businesses and information technology, there are other popular languages that were not chosen for this study such as C#, C, and Ruby (Dada et al., 2021). With different programming languages, the same method of calculating the Fibonacci sequence can be used, and other methods can be added as well to compare other factors of languages, such as comparing the different methods used to handle computer memory and comparing the CPU usage of each language.

This study may impact programmers who are unable to decide on a language to use on a project by allowing readers to decide on a language to use based on syntax, speed, and portability.

When beginning the research it was difficult to find a method used to compare programming languages without using a survey. While surveys allow for user opinion to be analyzed, the method of comparing language performance allows readers to learn more about

how a programming language works. This study allowed for a new insight on programming languages such as learning about popular languages and their common purposes in technology.

Summary

The purpose of this study is to compare three different programming languages in readability, efficiency in writing, speed, and portability. This study showed that Python had the best readability in relation to English, Python uses the least amount of code to accomplish a task, C++ uses the least amount of time to complete a task, and Java has the best portability and uses the most time to complete a task. This research allows for more information on different factors of programming languages. As technology continues to improve it is important to understand the reason why one language can be more preferred over others to use in a given situation.

References

- Adebola, Orogun. (2018). Introduction to Css. 10.13140/RG.2.2.28662.91200.
- Ali, Saqib & Qayyum, Sammar. (2021). A Pragmatic Comparison of Four Different Programming Languages. 10.14293/S2199-1006.1.SOR-PP5RV1O.v1.
- Aljas, Janjie & Amores, Honeyeth & Lincopinis, Darllaine. (2023). An Overview on C++ Programming Language.
https://www.researchgate.net/publication/371166631_An_Overview_on_C_Programming_Language
- Alomari, Halimi, Sivaprasad, Kaushik & Pandit, Zakaria & Halimi, Oualid & Sivaprasad, Kaushik & Pandit, Chitrang. (2015). Comparative Studies of Six Programming Languages.
- Beierlieb, Lukas & Bauer, André & Leppich, Robert & Iffländer, Lukas & Kounev, Samuel. (2023). Efficient Data Processing: Assessing the Performance of Different Programming Languages. 83-87. 10.1145/3578245.3584691.
- Cazzola, Walter & Olivares, Diego. (2016). Gradually Learning Programming Supported by a Growable Programming Language. IEEE Transactions on Emerging Topics in Computing. 4. 10.1109/TETC.2015.2446192.
- Chaw, Muhammad. (2023). Exploring Web Technologies: A Comprehensive Study on Apache Web Server, AJAX, Cookies, HTTP Sessions, and Web Development Languages.
https://www.researchgate.net/publication/380151866_Exploring_Web_Technologies_A_Comprehensive_Study_on_Apache_Web_Server_AJAX_Cookies_HTTP_Sessions_and_Web_Development_Languages
- Creswell, J. W., & Creswell, J. D. (2018). Research design: qualitative, quantitative, and mixed methods approaches. Fifth edition. Los Angeles, SAGE.
- Cutting, Vineesh & Stephen, Nehemiah. (2021). A Review on using Python as a Preferred Programming Language for Beginners. 8. 4258-4263.
- Dada, Oluwaseun & Aruleba, Kehinde & Yunusa, Abdullahi & Sanusi, Ismaila & Obaido, George. (2021). Information Technology roles and their most-used programming languages. 10.1007/978-3-030-96299-9_75
- Dr, Uday & Patkar, Uday. (2022). Python for Web Development. International Journal of Computer Science and Mobile Computing. 11. 36-48. 10.47760/ijcsmc.2022.v11i04.006.
- Đurđev, Darko. (2024). Popularity of programming languages. AIDASCO Reviews. 2. 24-29. 10.59783/aire.2024.59.
- Foster, Elvis. (2014). A Comparative Analysis of C++, Java, And Python Languages.
<https://www.elcfos.com/papers-in-cs/index/entry/id/8/title/Comparative-Analysis-of-C-Java-and-Python-Languages>

- Haris, Norhaidah & Hasim, Nurdattillah. (2019). PHP Frameworks Usability in Web Application Development. *International Journal of Recent Technology and Engineering*. 8. 10.35940/ijrte.C1020.1083S19.
- Huang, Xin. (2022). Introduction to Programming in Python. 10.1201/9780429155659-50.
- Islam, Md Tohidul & Islam, Rakibul & Jhilik, Rokshana & Islam, Asraful & Raihan, Prodhan & Faruque, Md & Md, Anik. (2024). A Comparative Analysis of Programming Language Preferences Among Computer Science and Non-Computer Science Students. *European Journal of Theoretical and Applied Sciences*. 2. 900-912. 10.59324/ejtas.2024.2(3).70.
- K P, Naveen Reddy & .Y, Geyavalli & .D, Sujani & Rajesh, More. (2018). Comparison of Programming Languages: Review. 9. 113-122.
- Liu, Zhiheng. (2023). Analysis of the application of Fibonacci sequence in different fields. *Theoretical and Natural Science*. 12. 167-171. 10.54254/2753-8818/12/20230459.
- Madan, Ankita & Kumar, Sanjay. (2012). Usability evaluation methods: a literature review *International Journal of Engineering Science and Technology*. CC By 4.0. https://www.researchgate.net/publication/266874640_Usability_evaluation_methods_a_literature_review
- Martinez, Desiree & Remegio, Axl & Lincopinis, Darllaine. (2023). A Review on Java Programming Language. https://www.researchgate.net/publication/371166744_A_Review_on_Java_Programming_Language
- Ogbuokiri, Blessing & Agu, Monica & B.O, Okwume. (2016). Comparison of python and java for use in instruction in first course in computer programming. *Transylvanian Review*. 24.
- Papadakis, Stamatios. (2019). Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: a comparison between Scratch and App Inventor. *International Journal of Teaching and Case Studies*. 10. 31. 10.1504/IJTCS.2019.10017717.
- Park, Thomas & Wiedenbeck, Susan. (2011). Learning web development: Challenges at an earlier stage of computing education. 125-132. 10.1145/2016911.2016937.
- Parveen, Zahida & Fatima, Nazish. (2016). Performance Comparison of Most Common High Level Programming Languages. *International Journal of Computing Sciences Research*. 5. 246-258.
- Pellet, Jean-Philippe & Dame, Amaury & Parriaux, Gabriel. (2019). How beginner-friendly is a programming language? A short analysis based on Java and Python examples. *ISSEP 2019 – Informatics in Schools: Situation, Evolution, and PerspectivesAt: Larnaca, Cyprus*
- Rijanandi, Teguh & Adhinata, Faisal. (2022). Choosing the Right Programming Language in Making a Website Backend Using the Waterfall Method. *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*. 10. 62-69. 10.3991/ijes.v10i02.30845.

- Saito, Daisuke & Yamaura, Tsuneo. (2013). A New Approach to Programming Language Education for Beginners with Top-Down Learning. *International Journal of Engineering Pedagogy*. 3. 10.3991/ijep.v3iS4.3216.
- Sears, A. (1995). AIDE: a step toward metric-based interface development tools. 101-110. Paper presented at Proceedings of the 1995 8th Annual Symposium on User Interface Software and Technology, UIST'95, Pittsburgh, PA, USA.
- Seffah, Ahmed & Donyaee, Mohammad & Kline, Rex & Padda, Harkirat. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal*. 14. 159-178. 10.1007/s11219-006-7600-8.
- Shukla, Abhishek. (2024). Cloud-Based Lightweight Modern Integrated Development Environments (IDEs) and their Future. *Journal of Artificial Intelligence & Cloud Computing*. 10.47363/JAICC/2024(3)218.
- Singh, Nehul & Singh, Satyendra & Verma, Karan. (2021). Object oriented programming: Concepts, limitations and application trends. 10.36227/techrxiv.16677259.
- Zhang, Hongyu. (2009). An Investigation of the Relationships between Lines of Code and Defects. *IEEE International Conference on Software Maintenance, ICSM*. 274-283. 10.1109/ICSM.2009.5306304.
- Zhang, Xian & Qu, Taiguo & Liu, Yinwen & Yin, Dong. (2020). Teaching Method Reform and Exploration of C Language Programming Training Course Based on Case. 10.2991/assehr.k.200723.096.