

A COMPARISON ANALYSIS BETWEEN THE C++ AND PYTHON PROGRAMMING LANGUAGES

Abdelatti A. Blg

Computer Science Department

Faculty of Science_ Ajaylat

Zawia University

a.blg@zu.edu.ly

Abdalla Mohamed Alasoud

Computer Department

Faculty of Science

Alasmarya Islamic University

ab.alasoud@asmarva.edu.ly

ABSTRACT

With the development of programming languages, we have noticed the dominance of C++ based in the programming field. In the past few years, we've also seen an increase in interest in Python. This article quotes the results of comparative research between C++ and Python programming languages. The criteria used for analysis are standard programming language functions and programming language principles. The paper advances the rationale for the study, an overview of each language. Moreover, this study will use two approaches. The first one about the emphasis is placed on expressions and assignment statements, and The second approach focuses on exception handling. Finally, this study is to investigate and compare the framework for both languages. The comparison will highlight areas where the current language provides broad support and which it lacks, and provide a summary.

Keywords: *Languages, Python, C++, Expressions, Exception.*

1. INTRODUCTION

Programming languages are a wonderful and interesting field of research. Computer scientists have created many new programming languages. In the past few years. Several different languages have been developed. Some languages are very popular, while others offer new features. Every language has its advantages and disadvantages. This study compared the different functions used by two popular programming languages which are C++ and Python. With these languages and their widespread use, programmers need to be aware of the advantages and disadvantages that each language can provide for their software solutions, and be careful when making reasonable decisions. These languages are compared in handling exceptions for

attributes, expressions, custom data, syntax, logic, and runtime errors. [11] It will use other criteria, such as implementing and running the same set of programs in all languages under study.

2. BACKGROUND (LITERATURE REVIEW)

The basic principle of the relative analysis between C ++ and Python programming languages is to create a better background for choosing the right language, enhance understanding of the importance of implementation, and improve the learning of current programming languages. Therefore, anyone with a career in software engineering will know how to choose the right language for the right project based on an understanding of programming principles.

One way to improve your ability to choose and learn programming languages is simple. Learning a first language is difficult. One of the problems is that you may be dealing with languages completely different from the others. Therefore, it will represent only a small portion of the basic concepts defined for modern programming languages. The second question involves investing time in researching new languages and how to use that language appropriately in specific programming. [12]

By comparing two programming languages based on basic concepts, we will be able to build a foundation around programming language paradigms and problems, to provide solid preparation for the effective selection and learning of new languages. The languages we'll be breaking down are C ++ and Python, in part due to their popularity. According to the April 2014 TIOBE Index, based on a comparison of ten popular programming languages, C ++ ranked 4th and Python ranked 8th. [12].

2.1 Overview of C++ and Significance of Language

Through the early 1970s, the C++ programming language was designed as a higher level version of C, providing support for object-oriented programming. And operating systems were written in assembler language. The C++ programming language was originally developed as a language to replace assembler in systems programming. C++ began to grow in popularity not just for systems programming, but as a general-purpose programming language. Its features like Virtual, violate, and Friend some of the very important OOPS features, rendering this language unworthy of being called completely Object Oriented. It's a middle-level language. The language is also widely used for teaching and research because it is clean enough for the successful teaching of basic concepts. C++ has greatly influenced many other popular programming languages. [3]

C++ is an object-oriented, statically typed, universal, purpose compiled, free form programming language that supports procedural, case sensitive, and generic programming. Reformed other languages, such as Python, virtual machines, and complicated run-time libraries have not traditionally been required for C++. Traditional development tools and compatibility with C++ libraries are emphasized.

The importance of the language is generally due to it being extremely fast, supporting a variety of different coding styles, and giving the programmer low-level access to the computer program through being a middle-level language. When dealing with large projects, the object-oriented principles. [11].

2.2 Overview of Python and Significance of Language

Through the late 1980s and early 1990s at the National Research Institute for Mathematics and Computer Science in the Netherlands discovered the Python Language and it was an interpreted, high-level and general-purpose programming language that was originally developed by Guido van Rossum.

Python is derivative from many other languages, including ABC, Modula-3, C++, Unix shell, Fortran, and other scripting languages. [2]

Python is one of the most popular dynamic programming languages currently, along with Perl, Tcl, Ruby, and RPH. While it is regularly viewed as a scripting language, it is a general-purpose programming language along the lines of Smalltalk or Lisp. Today, Python is used by the likes of experts writing applications for the world's fastest supercomputers to children learning it as a first programming language.

Python is the fourth the most used language according to Stack Overflow's 2020 survey, it's used by 44.1% of designers. It's also the third most loved language with 66.7% of developers admitting that it's their favorite language. [4]

Python's feature highlights:

- ✓ Easy it's simple to use and learn.
- ✓ Speed it speeds up development.
- ✓ Robustness you get production-ready code with automatic interactive documentation.
- ✓ Reduced bugs eliminate about 40% of human-generated errors.

- ✓ Short cuts code duplication to a minimum, which also results in fewer bugs.

Python is very important for software development as it gets many different things right, and in a combination that is not found in other languages. It classifies that language you'll spend a lot more of time reading code than writing it. By combining a powerful library with a user-friendly language, Python has been able to excel in a field with established giant languages like C++ and Java. And Python also recognizes that speed of development is important, so it emphasizes guiding developers to write readable code, and it has access to powerful constructs that avoid tedious repetition of code. [5]

3. METHODOLOGY

The principal aim of this investigation is to present and compare the important features and progressive features of two programming languages C++ and Python to establish favorability and prospect expectations of each language, concerning one another. The advanced features will cover are based on the following principles:

- ✓ Exception Handling
- ✓ Expressions and Assignment Statements

3.1 Exception Handling

Exceptions may refer to unexpected situations of user enters a character where an integer is expected and the program uses an array subscript that is outside of the range of values, additionally, an attempt is made at dividing by zero.[5]

There are three types of programming:

- ✓ Syntax errors

- ✓ Logic errors
- ✓ Runtime errors

In C++, exceptions are not handled hierarchically as done in Python. Instead, all exceptions are treated similarly. You can throw an exception from any function and catch an exception, report it, or re-throw it for another function. C++ also does not require any modification to its method definition as Java does, but may use the throw keyword deliberately, as depicted in figure 1. And C++ does utilize a try-catch, similar to Java. But, the whole exception handling mechanism is much simpler than in Python. For more discussion of C++ exception handling, please refer to Figure 1. [6] and [10]

```
const int DIV_BY_ZERO = 10; // A global constant
// ...
double divide (double First, double Second); // Function prototype
// ...

int main(int argc, char* argv[])
{
    // ...
    try
    {
        double Result;
        double FirstNumber;
        // ...
        Result = divide(FirstNumber, 0);
        // ...
    }

    catch (int ErrNum)
    {
        if (ErrNum == DIV_BY_ZERO) cout << "Cannot divide by zero!\n";
        // ...
    }

    // ...
} // End of main

double divide (double First, double Second)
{
    if (Second == 0) throw DIV_BY_ZERO;
    return First/Second;
}
// ...
```

Figure 1: Illustrating Exception Handling in C++ [3]

In Python, there are two significant features to handle the unexpected error. Standard exception handling. Python uses two syntaxes. Instead of the try-catch block, however, Python calls it a try-except block, also has an optional third-part, the final statement. In Python, this is an else clause. A simple example of the syntax is shown in figure 2.

The second is assertions, which are often placed by programmers at the start of a function to check for valid input, and after a function call to check for valid output.

```
try:
    Operations here;
    ...
except ExceptionI:
    If there is ExceptionI, then execute this block.
except ExceptionII:
    If there is ExceptionII, then execute this block.
else:
    If there is no exception then execute this block.
```

Figure 2: Python Try-Except-Else Block. [3]

3.2 Expressions and Assignment Statements

This section contains:

- ✓ Expressions
- ✓ Assignment Statements

3.2.1 Expressions

The syntax basic components of programming languages are in Datatypes. Variables and expressions. Expression was affected by the following factors: operator precedence, gender mismatch, data influence, and short circuit evaluation. Expressions are the building blocks for arithmetic and Boolean uses, as well as assignment statements. [5]

C++ allows the computer programmer to overload most of the operators by creating operator tasks so that they relate to complex objects. Indirectly, these approaches allowing handling of all objects as instances of the Object Class. [5]

Conditional expressions are used to evaluate a statement as true (1) or false (0), such as inside an if statement. The data type of the value returned by an expression depends on the elements used in the expression as examples of expressions showed in Figure 3 in bold.

```
int cadence = 0; // cadence represents the value 0
anArray[0] = 100; // anArray[0] represents the value 100
// The + operator concatenates the inside of the println to form one string value.
System.out.println("Element 1 at index 0: " + anArray[0]);

int result = 1 + 2; // arithmetic expression that represents the value 3
if (value == value2) // conditional expression that represents either 0 or non-zero
System.out.println("value1 == value2"); // no manipulation done is still an expression
```

Figure 3: Expression Examples [3]

Expressions are the same in C++ and Python, in the sense that they are all parts of a statement that result in a value.

3.2.2 Assignment Statements

Phrases are roughly equivalent to sentences in natural language. The statement forms a complete unit of execution, and the expression forms part of the execution. In Figure 4, the bold portion of the code is an expression, and each line of code is a statement. Therefore, these statements include things like variables and print commands.

In C-based languages (such as Java and C ++), the assignment clause has the following form: <variable> = <expression>. This statement changes the

variable value (assignment factor) on the left side of the equal sign to the value of the expression on the right side. Generally, variables are only defined with variable names, but some expressions define variables. Java treats tasks as expressions and instructions. What officially makes this a statement is the use of a semicolon. As an expression, its value is the value assigned to the variable. This is done to allow several assignments in a single statement, such as `a = b = 6;`

By giving `b = 6` as an expression with the value 6, Java types sense of this statement by assigning 6 to a, as well as 6 to b. This is an acronym for programming, which can improve or decrease readability and simplicity based on the point of view of the programmer. More shortcuts can be found in Using Compound Distribution Factors. The compound distribution factor is an operator that performs the calculation and distribution at the same time. (7) All 11 binary arithmetic operations for Java and C++ have equivalent compound assignment factors, as depicted in table 1.

TABLE 1: COMPOUND ASSIGNMENT OPERATORS IN JAVA AND C++

Operator	Description
<code>+=</code>	Addition and assignment
<code>-=</code>	Subtraction and assignment
<code>*=</code>	Multiplication and assignment
<code>/=</code>	Division and assignment
<code>%=</code>	Remainder and assignment
<code>>>=</code>	Signed right bit shift and assignment
<code><<=</code>	Signed left bit shift and assignment
<code>>>>=</code>	Unsigned right bit shift and assignment
<code>&=</code>	Bitwise AND and assignment
<code>^=</code>	Bitwise exclusive OR and assignment
<code> =</code>	Bitwise inclusive OR and assignment

This is explained in the Python documentation, the Assign statement is used to ``relink the name to the value and modify the attributes or elements of

the modifiable object. " Figure 4 shows the BNF codes used to illustrate the customization of Python. [6]

```

assignment_stmt ::= (<target_list> =) + (<expression_list> | <yield_expression>)
target_list ::= <target>({, <target>}) [,]
target ::= <identifier> | (<target_list> | "["<target_list>"]" | <attributeref> | <subscription> | <slicing>)
expression_list ::= <expression> ({, <expression>}) [,]
yield_expression ::= yield [<expression_list>]
attributeref ::= <primary>.<identifier>
primary ::= <atom> | <attributeref> | <subscription> | <call>
atom ::= <identifier> | <literal> | <enclosure>
subscription ::= <primary> [<expression_list>]
slicing ::= <simple_slicing> | <extended_slicing>
simple_slicing ::= <primary> "["<short_slice> "]"
extended_slicing ::= <primary> "["<slice_list> "]"
slice_list ::= <slice_item> ({, "<slice_item>"}), ["."]
slice_item ::= <expression> | <proper_slice> | <ellipsis>
proper_slice ::= <short_slice> | <long_slice>
short_slice ::= [<lower_bound>]: [<upper_bound>]
long_slice ::= <short_slice> : [<stride>]
lower_bound ::= <expression>
upper_bound ::= <expression>
stride ::= expression
ellipsis ::= ...
call ::= <primary> ({<argument_list>} | <expression> genexpr_for)
// Note: expression and argument_list are also defined at [Python, 2014e]

```

Figure 4: BNF Notation – Python Assignment Statement [6]

The Python language also uses only seven compound assignment operators, as opposed to the eleven mentioned in table 1 for Java and C++. The first five are the same as those in Java and C++. All seven operators for Python are depicted in table 2 below.

TABLE 2: COMPOUND ASSIGNMENT OPERATORS IN PYTHON

Operator	Description
+=	Addition and assignment
-=	Subtraction and assignment
*=	Multiplication and assignment
/=	Division and assignment
%=	Remainder and assignment
**=	exponent and assignment
//=	Floor division and assignment

Likewise, all three languages use the simple assignment operator as a single equal sign. [9]

4. RESULT AND DISCUSSION

After a thorough discussion of the important and advanced features of these programming languages, a designer should have a reasonable understanding of what differences to look for when considering any of these languages. Starting from choosing a language that fits the target domain, usually based on the appropriate model to programming standards such as code and Syntax, software engineering has a straight approach to compare languages. In our conversation, Python has proven to be a more readable and simpler language, by design as shown in table 3. Both languages also fit multiple domains and compilation, however, have slight differences that may a smaller amount useful for the computer programmer.

TABLE 3: PYTHON AND C++ SOME FACTOR COMPARATIVE

ELEMENT	C++	PYTHON
Code	C++ tends to have large lines of code.	Python has fewer lines of code.
Syntax	C++ has lots of predefined syntaxes and structure	Python is informal to remember almost similar to human language.
Compilation	C++ is pre; compiled.	Python uses an interpreter.
Speed	C++ is faster in speed as compared to python.	Python is slower since it uses an interpreter and also determines the data type at run time.
Garbage Collection	C++ doesn't support garbage collection.	Python supports garbage collection
Efficiency	Less clean and manageable in comparison to python	Easier to maintain, object-oriented and simpler to use

5. CONCLUSION

Python is better for beginners in terms of its easy-to-read code and simple syntax, and Python has fewer lines of code. Additionally, Python is a good option for web development (back-end), while C++ is not very popular in web development of any kind. Python is also the leading language for data analysis and machine learning. Although C++ can also be used for machine learning, it is not a good option. In terms of simplicity, Python is easier to use and has a strong support system, and the unofficial Python to remember is almost like human language. However, C++ has lots of predefined syntaxes and structure. After all, Python is an interpreted language, and it cannot be the same compiled language as C++.

REFERENCES

- [1] D. Arnold, M.A. Bond, Chilvers, and R. Taylor, Hector (2009): Distributed objects in Python, In Proceedings of the 4th International Python Conference.
- [2] [netguru, 2020] IBM. What Is Python? A Comprehensive Guide. <https://insights.stackoverflow.com/survey/2020>[6 February 2020].
- [3] [IBM, 2014] IBM. C/C++ Supported Operating Systems. <http://publib.boulder.ibm.com/infocenter/db2e/v9r1/index.jsp?topic=%2Fcom.ibm.db2e.doc%2Fdb2eap0202.html> [15 February 2020].
- [4] [Python, 2017] python for beginners. Keywords in Python. 2017. <http://www.pythonforbeginners.com/basics/keywords-in-python> [16 February 2020].
- [5] [Python, 2014a] Portable Python. What is Portable Python?. 2014. <http://portablepython.com/> [10 February 2020].
- [6] [Foster, 2014] Foster, Elvis C. Software Engineering: A Methodical Approach. New York: Apress Publishing, 2014. See chapter 18.
- [7] [IBM, 2014] IBM. C/C++ Supported Operating Systems. <http://publib.boulder.ibm.com/infocenter/db2e/v9r1/index.jsp?topic=%2Fcom.ibm.db2e.doc%2Fdb2eap0202.html> [16 February 2020].

- [8] Baduel, L.(2006) Programming, Languages, Deploying for the software. [online] available from <[citeseerx.ist.psu.edu/viewdoc /summary?doi=10.1.1.58.7806](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.7806)>[20 February 2020].
- [9] [C++, 2013] preference C++ keywords.2013. <http://en.cppreference.com/w/cpp/keyword> [20 February 2020].
- [10] Zakaria Alomari and Oualid El Halimi (2015): Comparative Studies of Six Programming Languages, research gate Article, Python Conference.
- [11] Karl Fast. (2003). What is "pythonic"?. Available: <https://mail.python.org/pipermail/tutor/2003-October/025932.html>. >[20 February 2020].
- [12] [TIOBE, 2015] TIOBE Software. TIOBE Index (2014). <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> >[13 February 2020].