

Project 1: Comparing C++, Python, and Java

MARISA LORAAS, New Mexico Tech, United States

ACM Reference Format:

Marisa Loraas. 2021. Project 1: Comparing C++, Python, and Java. 1, 1 (March 2021), 9 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

Author's address: Marisa Loraas, New Mexico Tech, Socorro, United States, marisa.loraas@student.nmt.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

XXXX-XXXX/2021/3-ART \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

Today, some of the most commonly used programming languages include C/C++, Python, and Java because they all bring different strengths and weaknesses within their features that make them ideal for different tasks/problems. This paper is meant to describe these three languages in depth in order to compare and contrast the three. Based on the characteristics that compose Java, Python, and C++, we will be able to understand each languages power, readability & maintenance, security, robustness, efficiency, and platform independence in comparison to other languages.

2 JAVA

2.1 Simplicity and Orthogonality

Java was designed to be very simple, as its designers strived to remove features of its ancestor languages (Ada, C, C++ etc) so that the number of language constructs that a programmer may need to know in order to produce a reliable applications is very minimal. However, it is a very verbous language which makes comprehension and readability harder. Lets look at a HelloWorld program to explain:

```
class HelloWorld
{
    public static void main(String[] args){
        System.out.println("Hello World");
    }
}
```

It's important to not that part of Java's simplicity is its familiarity; it was designed with the intention of resembling other coding languages so that a programmer could easily gain fluency in this language. However, this lies on the assumption that the programmer has experience, so in terms of simplicity this code may not be the simplest for a person who has never learned a coding language before.

For example, in our HelloWorld class: the class declaration, main function declarations, main function parameters, and the call to the System package just to print something may be very complicated for a new programmer but very easy for someone more familiar with programming.

2.2 Control Structures

Java includes various different control structures that effect the flow of the program. It has conditional branch control structures(sometimes called "Selection") such as if/else/else if statements and switch statements. Java also has loop control structures (sometimes called "Repetition" or "Iteration") like while loops, for loops, and do while loops. Finally, it has two branching statement control structures: break and continue.

2.3 Data Types and Structures

There are 8 primitive data types in Java: byte, short, int, long, float, double, char, and boolean. Everything else that is not a primitive data type in Java, is treated as an object. But even primitive types can be encapsulated into objects if the programmer needs.

2.4 Syntax Design

Java is designed so that its syntax must be in a very specific format and very verbose, which makes it a less intuitive compared to other languages. It's case sensitive, and their are specific rules to what you can capitalize and can't. All class names should begin with an upper case letter, if the

name consists of several words, each words beginning letter should also be capitalized. All methods should begin with a lower case. In Java, the program file name must also exactly match the class name. To define the main function in a Java program is also a very specific line of code. Its syntax also very heavily relies on curly braces and semicolons. For example, say we wanted to make our first Java class that called our first method, it might look something like this:

```
//Program name: MyFirstProgram.Java
public class MyFirstProgram{
    public static void myFirstMethod(){
        System.out.println("This is my first method!");
    }

    public static void main (String []args){
        myFirstMethod();
    }
}
```

2.5 Support for Abstraction (system and user level)

Java's Standard Library allows for a high level of abstraction, as it includes numerous packages that save the programmer implementation time because they don't have to use any time setting up specific data structures or operations.

2.6 Typing System

Java is what we would call a statically typed language i.e. a variable must be defined before it can be used. It's also strongly typed, which means every variable must be declared as a specific data type and cannot change type without a given conversion. It's typing system includes classes, primitive types, and other reference types that is related to a class in some way.

2.7 Exception Handling

In Java, all exception and errors are considered objects that are sub-classes of the class "Throwable". Exception Handling is handled by default by the run-time system when it receives an exception object. Java also allows customized exception handling via try-catch blocks.

2.8 Concurrent Facilities

Java includes a package of concurrency utilities that are used to build concurrent classes or applications from the the ground up. There are also high-level concurrency API's for Java.

2.9 Memory Garbage Collection

Java a built in automatic garbage collection in order to avoid memory leaks, which is one of the best aspects that this language has to offer.

2.10 Compiled, Interpreted, or Hybrid

Java is a Hybrid language, which means it is both compiled and interpreted.

3 C++

3.1 Simplicity and Orthogonality

C++ is probably the most complex of the three languages discussed in this document. Its lack of abstraction makes it so that the programmer must implement every attribute that the computer needs to understand this code. This leads to lengthy and more complex programs compared to other languages. Here's a HelloWorld program in C++:

```
//HelloWorld.cpp
#include <iostream>

int main(){
    std::cout << "Hello World!";
    return 0;
}
```

Even from looking at as, we are the library invocation and the variable declaration show us how illogical this language is as far as intuitiveness for syntax goes. This makes the language very difficult to learn and read, especially for people with no programming experience.

3.2 Control Structures

C++ has various different control structures that effect the control flow of its programs. It has conditional control structures like if/else if/else statements and switch statements. C++ has loop control structures like while loops, for loops, and do while loops (which are included in every language in this document). C++ also has control structures that are referred to as Jump statements, these include: break, continue, and goto statements and exit function. Notice that the goto statement is not included in the other languages observed in the document.

3.3 Data Types and Structures

These are the primitive data types of C++: bool, char, int, float, double, void, and wchar_t. Beyond these primitive data types there are various other structures in c++ like enumerations, pointers, arrays, references, and classes.

3.4 Syntax Design

Like Java, C++'s syntax relies very heavily on the use of curly braces and semicolons. It also has the same naming conventions as Java, using CamelCase for function names and program names. Though not as verbose as Java, C++'s syntax is still pretty complex not very intuitive. For example, a C++ program for someones first program and first call to a function may look like so:

```
\\MyFirstProgram.cpp
#include <iostream>

void myFirstFunction(void);

int main(){
    myFirstFunction();
    return 0;
}

void myFirstFunction( void ) {
```

```
std::cout<< "This is my first function!";
}
```

3.5 Support for Abstraction (system and user level)

C++ probably has the lowest level of abstraction out of all of the languages in this document, as it has very limited library support which forces to the programmer to implement functions that work directly with machine code.

3.6 Typing System

C++, just like Java, is a statically-typed language and a strongly typed language. This means the programmer must specify the type when declaring a variable or instruct the compiler to deduce the variables type for the programmer.

3.7 Exception Handling

C++ handles exceptions with try-catch blocks (just like in Java and Python) and also the "throw" keyword is used by to programmer to throw a specific exception.

3.8 Concurrent Facilities

C++ originally could not support concurrency. However, newer updates to C++ have allowed for the support of multi-threaded programs, which are and can be used for concurrency.

3.9 Memory Garbage Collection

C++, unlike the other two languages in this paper, does not have any sort of implicit garbage collector, which means that memory garbage collection must be implemented by the user, which can be difficult.

3.10 Compiled, Interpreted, or Hybrid

C++ is a compiled language, which means C++ code is converted into machine code that the processor can execute.

4 PYTHON

4.1 Simplicity and Orthogonality

Python is probably the simplest language out of the 3 described in this document. It has clear syntax, very few lines of code(due to its powerful dynamic semantics), and very closely mimics the English language so its very intuitive for programmers of all levels.

```
#HelloWorld.py
print("Hello World")
```

As we can see, this program is pretty simple to anyone who looks at it and is only one line of code (the shortest number of lines out of the three). We can see from the *print* statement that this language is very similar to how we describe this problem in English: "Make a program that prints "Hello World"". Which makes it extremely simple to learn, read, understand, and program with.

4.2 Control Structures

In Python, the flow of control of the program is implemented by three types of control structures. Sequential control structures are when sequential statements are executed one after the other,

which doesn't require any effort on the programmers end in python. It has conditional branch control structures like if/else/if else statements and switch statements. Python also has looping control structures such as while loops, do while loops, and for loops.

4.3 Data Types and Structures

These are the 12 primitive types defined in python: byte, int, float, long, complex, bool, str, list, tuple, range, set, frozen set, bytearray, memoryview, and dict. Python does not have a char data type. It's also important to note that Python considers lists, sets, and dictionaries as primitive types, not advanced data types like many other languages do.

4.4 Syntax Design

Python was designed to be a very readable language, so its syntax aligns with English which makes its very easy to read and write python code compared to the other languages discussed in this document. Where Java relies on curly braces, Python's syntax heavily relies on indentation. It's naming conventions rely more on underscores than CamelCase (like Java and C++ do). For example, say we wanted to make our first python program with our first Python function, it may look something like this:

```
#MyFirstProgram.py
def my_first_function():
    print("This is my first function!")

my_first_class()
```

4.5 Support for Abstraction (system and user level)

Out of the three languages discussed in this document, Python allows for the highest level of abstraction. Its standard library is extremely expansive and is open source, which means that the community may contribute to it.

4.6 Typing System

Python is a dynamically typed language, which means that type checking is done at run-time and the type of a variable is allowed to change over its lifetime. This allows for Python to be a lot more flexible than the other languages in this document. Python is also strongly typed, which means that the value of a type doesn't change unexpectedly; it requires a explicit conversion.

4.7 Exception Handling

Python exceptions are handled by implementing try and except blocks. Python also allows users to raise or even make their own exceptions.

4.8 Concurrent Facilities

Python 2 and 3 have numerous API's that are dedicated for concurrent programming. Python also has modules for concurrent execution of code, but this depends on an event being IO or CPU bound and it having preemptive or cooperative multitasking.

4.9 Memory Garbage Collection

Python has a built in automatic memory garbage collector that effects memory and garbage handling without unnecessary memory leaks. It also allows for programmers to perform manual garbage collection.

4.10 Compiled, Interpreted, or Hybrid

Python is an Interpreted language, which means Python code is being executed by another program and not the target machine.

5 POWER

Python is probably the most powerful language when in comparison to C++ and Java (and arguably most other languages). It's definitely has the shortest syntax of the three, which can be seen within the code block examples given previously, and its semantics are extremely powerful. It has an massive and expansive library, and is extremely easy to implement and learn. Java also has a very large standard library and power semantics but is just too verbose to compete with Python. C++ by far the least powerful of the three, as its characteristics and standard library cannot compare the power we get from Java and Python.

6 READABILITY AND MAINTENANCE

Python is definitely the most readable and most easy to maintain of the three, as its syntax closely resembles English which allows it to be readable to people who may not be considered programmer's as well. Java is also considered to be very readable and easy to learn; its verbosity just does not make it as readable as Python. C++ is definitely the most complicated when it comes to implementation and syntax, which makes it just not as readable or easy to maintain as Java and Python.

7 SECURITY

Java and Python are both fairly equal in terms of security: they have minimal vulnerabilities and their automatic memory garbage management allows for the languages to prevent users from doing anything unintended. C++ is only just below these two because of its lack of memory garbage collection.

8 ROBUSTNESS

All three languages seem to be fairly robust, as they have nearly the same ways of dealing with errors and exceptions very efficiently but also effectively.

9 EFFICIENCY

C++ is the most efficient out of our three languages discussed here. Because it is a statically and strongly typed language, but it also a compiled language, it can most efficiently be compiled into machine code and executed. Java is close to C++, because they are both strongly and statically typed, but less efficient because it is a Hybrid language. Python is the least efficient because it is an interpreted language and dynamically typed.

10 PLATFORM INDEPENDENCE

Platform Independence (also referred to as portability) applies most to Java, as Java code can be interpreted by the JVM from any machine because of its Hybrid nature. Python is also platform independent, though its bytecode is not platform independent like Java's is. Finally, C++ is the least platform independent because it cannot be recompiled on different platforms.

11 REFERENCES

Anon

. 2018. Exceptions in Java. (September 2018). Retrieved March 15, 2021 from <https://www.geeksforgeeks.org/exceptions-in-java/>

Anon

. C++ Exception Handling. Retrieved March 15, 2021 from https://www.tutorialspoint.com/cplusplus/cpp_exceptions_handling.htm

Anon

. C++ Variable Types. Retrieved March 15, 2021 from https://www.tutorialspoint.com/cplusplus/cpp_variable_types.htm

Anon

. Code Style. Retrieved March 15, 2021 from <https://docs.python-guide.org/writing/style/> Anon. Concurrency in Python - Quick Guide. Retrieved March 15, 2021 from https://www.tutorialspoint.com/concurrency_in_python/concurrency_in_python_quick_guide.htm

Anon

. Concurrency Utilities Overview. Retrieved March 15, 2021 from <https://docs.oracle.com/javase/7/docs/technotes/guides/concurrency/overview.html>

Anon

. How Java Garbage Collection Works. Retrieved March 15, 2021 from <https://www.dynatrace.com/resources/ebooks/javabook/how-garbage-collection-works/>

Anon

. Python Control Structures. Retrieved March 15, 2021 from <http://net-informations.com/python/flow/default.htm>

Anon

. Structure of a program. Retrieved March 15, 2021 from https://www.cplusplus.com/doc/tutorial/program_structure/

Anon

. Retrieved March 15, 2021 from https://www.w3schools.com/python/python_datatypes.asp

Anthony D. Williams

. 2012. C++ Concurrency In Action, Shelter Island, New York: Manning Publications. Corob-Msft. C++ type system. Retrieved March 15, 2021 from <https://docs.microsoft.com/en-us/cpp/cpp/cpp-type-system-modern-cpp?view=msvc-160> David Flanagan and Ben Evans. 2018. Java in a Nutshell 7th ed., New York, New York: O'Reilly Media.

Geir Arne Hjelle

. 2021. Python Type Checking (Guide). (January 2021). Retrieved March 15, 2021 from <https://realpython.com/python-type-checking/#type-systems>

Guest Contributor

. Basics of Memory Management in Python. Retrieved March 15, 2021 from <https://stackabuse.com/basics-of-memory-management-in-python/>

James Gosling and Henry McGilton

. 1996.(April 1996). Retrieved March 15, 2021 from <https://www.oracle.com/java/technologies/simple-familiar.html>

James Gosling et al.

2021. The Java Language Specification. Java SE 16 Edition (February 2021), 1–844. DOI:<http://dx.doi.org/https://docs.oracle.com/javase/specs/jls/se16/jls16.pdf>

N.A. Baeldung

. 2019. Control Structures in Java. (October 2019). Retrieved March 15, 2021 from <https://www.baeldung.com/java-control-structures>

STXNext

Sp.z.o.o.Retrieved March 15, 2021 from <https://www.stxnext.com/python-vs-other-programming-languages/>