Programming Languages – COMP712

Assignment 1 – Comparing Two programming languages & their issues

Nikkolas Diehl

Comparing C++ and Python

**Python**:

Python was created in 1991 by Guido van Rossum (King, 2019). Python was built as a general-purpose programming language with simplicity as its backbone.

**C++**:

C++ was created in 1979 by Bjarne Stroustrup. C++ is designed to take all the best features of C and make it an OO (object orientated) programming language as a general-purpose programming language more built for adaptivity and design over simplicity. (King, 2019)

**Five Key Differences**:

1. Memory Leakage

(Differences Between Python vs C++, n.d.) (Muphy, 2012). Whilst python has automatic and inbuilt garbage collection and dynamic memory allocation for arrays, variables and classes, C++ does not have an inbuilt memory collection and the developer must work to make sure memory does not leak.

C++ encourages the use of pointers and references to save on memory. When sharing variables between classes, for example; C++ can share a direct reference through a function. This direct access of memory does allow a lot more control over your program but has glaring issues with memory leaking and empty pointers.

Python on the other hand has automatic allocation and deallocation of memory and variables. A developer will not have to pre-allocate or specify a deconstruction of data due to Python using dynamic memory allocation. Python does this by using memory allocation referencing counting. (Python Garbage Collection, 2018)

1. Type Binding and Variable Declaration

C++ Types are explicitly declared. Bound by a type name such as bool, std::string, int, float etc. Types are checked at the compilation time and are strict. Python types on the other hand are bound instead to just the values checked at run time by an interpreter and are dynamic rather than strict. Python also uses its interpreter and simplicity to make sure the program is safe by checking a variable can never be anything but a pre-defined type. This difference reduces complexity in Python and optimises the programmer efficiency.

Also, Python has dynamic type casting and type conversion. In Python you can write the following code:

a = 3;

b = 7;

c = b/a;

And this will output a floating-point number automatically whereas in C++ the variable type must be declared and explicit type casting is needed to convert the two integer variables to a floating-point variable.

1. Language Complexity:

Python is well known for its very readable code and well documented code base. It is a very high language and as such uses an interpreter rather than just a compiler allowing a much faster work flow as the programmer does not have to worry about the static programming style from C++. C++ has about 800+ pages of documentation for its language (Muphy, 2012) and has to import a lot of external API’s and libraries to add simple features. Although because Python uses an interpreter and is a higher-level language, this cost run time speed and means C++ is actually faster at run time and thus is usually better for most high cost, large scale performance-based applications such as game development. (Rohner, 2018). Because of complexity issues with C++, an application can take over a year when made in C++ can take only a few months in Python.

1. Interpreted vs Complied:

C++ is explicitly pre-compiled for each individual operating system and situation to be able to run a most optimal speed. This causes issues when trying to do cross platform programming and is a slow and costly process for some applications. This also holds back rapid testing and exploration.

Python on the other hand was built with a “write once, run anywhere” idea allowing python code to dynamically run on any operating system with python installed giving developers the ability to rapidly test and update an application. Python’s interpreter also gives it the unique power to be used as a web development programming language as well. Another advantage of python using an interpreter is that python can act as a scripting language as well for AI/machine learning development as well as an application development language. (Rohner, 2018)

1. Variable Scope & Function explicitly:

In C++ variables only exist in memory within a specified scope. They can either be created and destroyed within class, function or single loop or defined in a header file across the entire C++ class. This can cause problems as pre-defining variables takes up memory, but creating variables within tight scopes means either saving a copy of that variable or outputting the variable in some way. Python however has dynamic variable access and is not limited by variable scope. Python is also object orientated so variable scope being a non-issue within the code base means much easier execution and optimised creation of an OO application.

# References

*C++ vs Java vs Python a comparison*. (2018, November 17). Retrieved from Javatips.net: https://www.javatips.net/blog/c-vs-java-vs-python-a-comparison

*Differences Between Python vs C++*. (n.d.). Retrieved from EDUCBA: https://www.educba.com/python-vs-c-plus-plus/

King, R. (2019, July 31). *Python VS C++: Let's Compare*. Retrieved from bitdegree.org: https://www.bitdegree.org/tutorials/python-vs-c-plus-plus/

Muphy, E. (2012, April 15). *What are the advantages of Python over C++?* Retrieved from Quora: https://www.quora.com/What-are-the-advantages-of-Python-over-C++

*Python Garbage Collection*. (2018, June 11). Retrieved from Digi International Inc.: https://www.digi.com/resources/documentation/digidocs/90001537/references/r\_python\_garbage\_coll.htm

Rohner, T. (2018, September 24). *Python Vs. C++ for Machine Learning - Language Comparison*. Retrieved from netguru: https://www.netguru.com/blog/python-vs.-c-for-machine-learning-language-comparison