

C++ is a general purpose object oriented programming language. C++ was developed by the Danish Computer Scientist, Bjarne Stroustrup in the mid 1980s. It was developed as an extension to the programming language C, which is heavily used for system programming. Python is an interpreted, high level general purpose object oriented programming language developed by a Dutch programmer Guido van Rossum in the early 1990s.

Even though these programming languages are general purpose and object oriented, they do share similarities and differences. One of the major differences between Python and C++ is that Python is much more readable. All variable in Python are objects, and therefore do not require the user to provide a data type during declaration. Python variables determine their data type by looking at the data that is being stored in it. C++ on the other hand is a lower level language compared to Python, so it is more syntactically heavier than Python.

As mentioned above, C++ was developed as an extension to the C language. C++ was originally to be called “C with Classes”, since one of the major aspect that C++ introduced to C was the capability for object oriented programming. Therefore C++ is backwards compatible with C, which means most non-trivial C programs will be compatible with C++ with modification. This makes programming in C much easier, for example, “string” data type does not exist in C, and users need to use array of characters to work with strings, however, you can very easily change the extension of the program file to “.cpp” from “.c” and be able to use the data type “string” by simply including the following line of code, “#include <string>”.

Python provides a the ability to “write once, run anywhere” which enables it to run on any operating system that Python installed. When it comes to compiling C++ programs, some syntax are operating systems specific, for example, if a C++ program has the following line of code, “system(‘pause’)”, it will throw an error when attempting to compile the program in a Linux operating system. Moreover, when using Visual Studio, attempting to take an user input ‘x’ and declaring a static array of size ‘x’ will have a compiler error, because the size of a static array should be specified at compile time. However when compiling the same program in a Linux distribution, it will not give an compiling error. Therefore not all C++ programs can be compiled and run in all operating systems without making some modifications to the code. Python does not have that issue, if a python program runs on Windows, it will also run on a Linux distribution.

Moreover, C++ does now provide garbage collection and uses pointers to a large extent, which is why C++ is prone to memory leak. For example, if a linked list program is written in C++, and thousands of elements are stored in the list, and it does not properly deallocate the memory that was declared in the heap for those elements, it can lead to major memory leak, and require a reboot of the system to be able to use it properly again. Unlike C++, Python has built in garbage collection, and dynamic memory allocation process, which is why Python has efficient memory management.

It doesn’t matter if it is Microsoft Windows or Mac OSX or Linux, they are all programmed in C++. C and C++ is the backbone of all major operating systems, owing to the fact that it is strongly typed and a fast programming language, this makes C++ an ideal choice for developing an operating system. C is one of those languages that is very close to assembly language, which also helps in writing lower level operating system modules.

Various web browsers’ rendering engines are programmed in C++ due to the speed that it offers. These rendering engines’ fast execution time decreases user’s waiting time for content to show up on the screen.

Moreover, C++ is heavily used for graphics because of their speed (fast rendering). Because of the speed that C++ offers, most of the video games that uses heavy graphics use C++ as their primary language. This helps reaching more customers because an optimized application can be run on devices that do not have high computational power.

C++/C is used for embedded systems, databases, cloud/distributed systems, machine learning, banking applications, telephone switches, compilers and many others. Because C++/C are used for designing and developing compilers because C++/C are relatively lower level languages that are closer to the hardware than any other languages but assembly, which makes C++/C ideal for developing compilers.

Python is used for web and internet development, desktop GUI applications, scientific and numeric applications, software development application, education, business, database access, network programming, games and, last but not least, 3-dimensional graphics.

Citations

“Applications for Python.” *Python.org*, www.python.org/about/apps/.

Goel, Aman. “C++ Language: Features, Uses, Applications & Advantages - Hackr Blog.” *Hackr.io Blog*, Hackr.io., 2 Jan. 2019, hackr.io/blog/features-uses-applications-of-c-plus-plus-language.

“Python vs C++ - Find Out The 9 Important Differences.” *EDUCBA*, 25 Feb. 2019, www.educba.com/python-vs-c-plus-plus/.

Radcliffe, Tom RadcliffeTom. “Python vs. C/C++ in Embedded Systems.” *ActiveState*, 3 Feb. 2019, www.activestate.com/blog/python-vs-cc-embedded-systems/.