Homework 4

For homework 4 we are asked to compare and contract C++ with a language we already know. And honestly the only one I studied before starting the post-bacc program was Python so that is the language I am picking to go into the differences about between C++. Python was invented 30 years ago in 1991 (which as an aside I was born in 1992 and looking this up makes me feel very old all of a sudden). Python is a multiparadigm language and drew influences from several languages; one of which was C++. However, with a big focus on clean design it differs from C++ in many ways as well.

One major difference is that python is an interpreted language and C++ is a compiled language. With interpreted languages like python all you have to do is type what you want and hit run. The computer figures out how to run the file directly from the source code. This is good for small things. But for C++ it must be compiled. While compiled a language is turned into machine code which optimizes the source code into an executable file. Another big difference is declaration. With dynamic languages like Python, if you want to declare an integer for example in Python, all you need to do I give a variable name and integer: x = 420. But in C++ the computer cannot simply glean from syntax what “420” is supposed to be. So the data type must be declared: int x = 420. A major difference between a language like Python and a language like C++ is pointers. For those of us who learned on Python and then moved to a language like C or C++ etc., pointers were a tough concept to understand. Pointers point to an address in memory to tell us where data is located. You can get the memory address in Python but you don’t really ever use pointers for anything in the language even if you can access them. However, in C++ you can use pointers to point to certain variables in memory without ever accessing the data.

Despite differences many programing languages have a lot of similarities regardless of certain syntax differences. One common thing in multiple programming languages including Python and C++ is the use of functions. Functions are pivotal to programming, as they help clean code practices. For instance, if you have a large chunk of code designed take a name, convert each character into ascii code, put the code in numerical order, then convert that back to letters to display the word in alphabetical order, that would technically be possible but pretty ugly and unclear what your are looking at In other languages you have the ability to create functions to do each one of these things and in a main function call these functions in order to do the same thing in a clearer way. Loops are another major part of most programming languages. Again, although syntax differs, they have the same function. For instance, if you want to create a while loop for a certain number of iterations, you can initialize a variable outside of a loop, set the lop to run while the variable differs from a desired target, do an action, then increment/decrement the variable. Finally, across languages they support the ability to take input and output. Interacting with users is a major part of a programming language so it would be hard to do anything without these common features.

In summation, the world of programming is vast. Trying to paint everything with a broad brush is well a pretty impossible task. Syntax for instance can vary wildly from language to language, and certain languages can hold different functionalities. However at their core languages also can serve similar purposes such as user interaction and logical operations.