**Basic Syntax of C++:**

* Must include the following:
  + ***#include <iostream>***
  + ***using namespace std;***
  + ***int main(){}***
* To put output into the main console we can use ***cout function*** (short for console output)
* ***endl function*** is the same as “\n” (short for endline)
* We must have have swirled parenthesis ‘{}’ to group executions.
* All lines of code must end in ‘;’ semi colon.
* At the start of all source codes, there is ***#include <iostream>***, this is also a method to include standard libraries in source code. Another example is ***#include <string.h> (c++ equivalent is #include <cstring>***
* The ***return 0;*** at the end of source code tells the program to terminate the main() function.
* Namespace: A syntactic container providing a context for names so that the same name can reside in different namespaces without ambiguity. Ie) modules functions, classes, methods etc.

**Basic Input in C++:**

* Coding basic string from users:
  + Define variable ***string input = “ ”;***
  + cout << “ string ”;
  + cin (short for console input) >> variable (input);
  + cout << “ string ” << input << endl;
  + Although this show only the the first word of a string that was input, might need to specific ***char[50]*** as an example.
* Another way to do this is:
  + Define variable ***string input = “ ”;***
  + cout << “ string ”;
  + ***getline(cin, input);*** (get line function from console input (cin) and variable assign to “input”.
  + cout << “ string ” << input << endl;

**String:**

* String functions must include ***#include <cstring>*** this is the string library for c++.
* ***strcat(variable, string),***this concatenates two strings specified
* ***strcpy(variable, string),*** copies string
* There is a string length function for counting the string accessed with the dot notation, ie  ***variable.length();***
* The C-style character string originated within the C language and continues to be supported within C++. It is a one-dimensional array of characters which is terminated by the null character “\0”. **Null-terminated string** contains the characters that comprise the string followed by a null to terminate the string.
* When creating array of strings in C++ we use single quote marks. If we use double quotes, it comes with error **[Error] too many initializers for ‘char[6]’**.
* We must have one element initializer to validate the string array created. The value can be larger than the exact length of characters but it cannot be smaller than the length of all characters. *Ie) char str[6] = {'H','e','l','l','o','!'};*
* Indexes can be accessed when the string variable is specified as character.

**Function:**

* The basic syntax for defining a function is :
  + *Return\_type function\_name(parameter list):*

{

Body of function

}

* Sometimes functions perform the desired operation but doesn’t return a value, in this case we use the keyword ***void***.
* When specifying a Boolean function, it returns 1 if the result is true or 0 if result is false, unlike Python