



**Game Engine Creation**

COSE40638

Student Name: Bruno Reeve

Student Number: r019574m

**Portfolio**

Contents

Chapter 1: Introduction

Chapter 2: Data Types

Chapter 3: Variables

Chapter 4: Operators

Chapter 5: Conditionals

Chapter 6: Loops

Chapter 7: Functions

Chapter 8: Arrays and vectors

Chapter 9: Strings

Chapter 10: Debugging

Chapter 11: Pointers

Chapter 12: References

Chapter 13: File Handling

Chapter 14: Object Orientated Programming – Structs

Chapter 15 OOP - Classes

Chapter 1: Introduction

To begin we will be going through the steps required to build a simple application in C++ using Visual Studio. This will be the traditional ‘Hello World’ program written by all new coders. This will go through the steps required to create a new program and how to add source files. If you followed along with the code-along then this section will just need a screenshot of your output.

NOTE: There is one function in the HelloWorld program. It is called main(). All programs must have a main function as this is where execution of the program begins. There can only be 1 main function in any given program.

Firstly, create a folder in your OneDrive and call it GitHub. Inside this create another folder called GEC. It is recommended you add folders for each week to make finding work easier, but this can be done later as needed.

**Program 1: Hello World**

1. To begin, start Visual Studio and you will be met with this window…

1. Click Create a new project and select the C++ Empty Project and click **Next**.
2. Name your Project HelloWorld and save to the location of your GEC folder within OneDrive that you created earlier, then click **Create**.
3. You should have the following window open. Right click the folder **Source Files** -> **Add** -> **New Item**. The default can be left as it is, it should have C++ File selected with the name **Source.cpp**. Click **Add** 
4. Now for some code! Replicate the following code: 
5. The first line of code adds the input/ output streams library that contains the cout object that allows us to print to screen. As mentioned above, every program needs a main function as a point of entry. As you can see this function is of type **int**, this means that the function expects a return type of type int, as seen on line 7 where we return 0. This return is reached only if the function is completed. On line 5 we start with **std::** this allows us to use the objects with the std namespace of the standard library, the use of **std::** can be omitted if **using namespace std;** is added below our include, but for now lets use best practice and include them. After cout we have two left chevrons << when outputting these will face left, putting what is on the right of them to the screen. If we were taking information in from a user, they would face right. We will cover this later. We then end with **endl;** which is the same if we were to add \n for a new line and of course the semicolon which is needed to end a line of code. Keep an eye out for these as they are a common result for program errors from new coders.
6. Now run your program using the green play button for Local Windows Debugger or F5. If you have followed along correctly you will be met with a console window containing “Hello World!”.
7. Screenshot this and add it below.

**Program 1 Screenshot:**