

Lab 8 – 08227 Advanced Programming

This tutorial introduces the reader to using classes in C++.

The content of this lab tutorial may be covered in the lecture after the scheduled lab, depending on timetabling. If this is the case, then try to complete as much as you can before the lecture, and then complete the rest of this lab next week during the next lab session.

1.0 Class Exercise 1

Copy your solution from Lab 7 to the folder **G:/08227/Lab8/**.

C++ is an object orientated language, so let's create a new object that will handle the file copy process for us.

In Visual Studio, right-click on the **Project Name** (Parser) then **Add -> Class** from the menu. Select **C++ Class** and click **Add**. Enter **FileUtilities** as the Class Name and click **Finish**. You should now have files **FileUtilities.cpp** and **FileUtilities.h**.

FileUtilities.h contains the class and function declarations (in this case just the default constructor and destructor) whilst **FileUtilities.cpp** contains the function code.

In your **main()** function create a new instance of **FileUtilities** (e.g. `FileUtilities fileUtil`), **remember** to include the header file. Compile your code.

Add the **Copy(char filenamein[], char filenameout[])** function from your **main.cpp** file to your **FileUtilities** object and rename this function to **textFileCopy(char filenamein[], char filenameout[])**. Remember to add the necessary header declarations and make sure that you make the **textFileCopy()** function a part of the **FileUtilities** class (i.e. add **FileUtilities::** before the **textFileCopy()** definition). In your **main()** function call the new **textFileCopy()** function. Compile your code.

Test your new **textFileCopy()** function by using it to copy a text file.

2.0 Class Exercise 2

Download the file **ObjectParserInput.txt** from the module SharePoint site. The text file contains a sequence of nested objects, where each object is delimited by characters **{** and **}**.

Attributes within the objects consist of a name followed by at least one white space followed by a set of parameters delimited by characters **(** and **)**. e.g. **colour (5)**.

A comma is used to separate multiple parameters e.g. **position (3,5,8)**.

The full set of named attributes are:

- colour (integer)
- position (integer, integer)
- size (integer)

You are to create a function in the **FileUtilities** class called **ObjectParser(char filenamein[], char filenameout[])** that will parse the input object file and:

1. Copy the data to an output file. ADVANCED: format the output so that each nested object is indented.
2. Locate the object with the largest size and print out to the console window the level within the object hierarchy. Assuming that the initial outer most object is level 1, your final printed text should state "Largest object is of size **14** and is located at level **4**".