Task 2 - Lab: C++ for Programmers

Summary:
You should all be programmers, but you may not be familiar or confident with C++ yet! If you are new to C/C++ programming, but you already have a strong programming language, work on finding the equivalent to what you already know. Even if you are confident, use this lab to quickly review your C++ skills so that you are ready for later work, and have good resources on hand to help yourself.

At the end of this lab you should have a working demonstration program for key C++ concepts, and a list of a number of good C/C++ references and resources.

What you need to do:

1. **Update** Doubtfire status for this task to "Working On It".

2. **Internet C++ resources.** As programmers we become familiar with good online resources when looking for answers or trying to solve problems. Do a deliberate search to find resources for the following, but don't take too long. (This is intended to be of real value later!)
    a. C++ quick reference / "cheat-sheet" documents. Watch out for different (old) C++ versions.
    b. Keyboard short-cut reference sheet for your code editor of choice.
    c. Online code debugging/testing tools that you could use to quickly test code.

    Save any resources that you download (pdf's etc) to your repository (provided you have permission), but make sure you note where they came from. Note links to resources, and why they are good, in your simple lab report document.

3. **Create a simple report.** Write a lab report (simple text document is fine, again like release notes for the previous lab) that states what you have done for this lab. Include your name, student id, the unit code, the task number and the date. Include in your report the resource links you found for point 1 above.

4. **Create a demo program in C++.** On the next page we have listed a number of C/C++ concepts and features we want you to demonstrate in a single, relatively simple program. Think of this as a tune-up check-list if you are quite familiar with C++ already. Store this in your repository in the appropriate folder.

5. **Update Doubtfire status, show your tutor:** When your repository is ready, and your simple text file "release notes" style report has been uploaded to Doubtfire, update your Doubtfire status and show your tutor.

Recommendations:

- Don't get stuck on little things. If you get stuck, ask your tutor and other students as you go, or try moving on to other parts and return to problem topics later.

Lab Demo Code Features

Your simple C++ program must demonstrate the following features. Many are simple C features. You can implement them in any order you want, but refer to them by number so it is easier to find things.

1. Create a function that accepts at least two values (simple parameter values, not user input), prints the values to screen, and returns nothing. Call this function from your main.

2. Create a function that accepts one value, alters it, and returns it (simple value). Call this function from main, and print the returned value.

3. Declare a pointer to a variable, and demonstrate the setting and reading of the value in the variable using the pointer.

4. Create a function that uses a for loop to print all odd numbers between 0 and 20 on a single text line of text output, formatted with spaces. Call this from main.

5. Create a function that creates a 1-dimensional array of 5 integers, then fill the array with random int values, print the contents of the array to the screen. Call from main.

6. Split the line of text, such as “this has spaces in it”, by spaces, and show each part to screen.

7. Declare a simple class that has at least one private variable, one public variable, and one public method. Create an instance of the class, and demonstrate its use.