Lab 2 C++

Deadline: 26th October @5.pm

Your name and student number must be at the top of every file.

All the arrays are of **int** or **char** datatype. You cannot use <string>or anything else.

All code must be consistently indented and variables given appropriate names.

To Upload:

Code (cpp file)

Lab Book (pdf or doc)

Link to demo video. ( put in comments beside your name on top of cpp file.)

You must upload the code to blackboard before 26th October @5.pm even if not all the questions are completed.

If no code is uploaded by the deadline, you get zero for the lab.

If no demo video link is uploaded by the deadline you get zero for the lab.

If no Lab book is uploaded by the deadline you get zero for the lab.

**You must upload a video link for the**  the lab and **upload the code** (and **Lab Book doc )**to get a mark.

Demos will be picked on a random basis also. You may not be picked for demo.

You must create a **4 or 5 minute video** explaining the code and how you came about the solution of 3 or 4 of the more difficult questions below.

You can use ObsStudio (https://obsproject.com/ ) to screen record. You should use your rough work and code to explain how you came about your solutions.

Contents of Video

No more than 5 minutes!

Running the code from visual Studio.

Pausing on the googletest results.

Explain how you came about your code solution, supported with your code and your lab book. It may include approaches you tried but didn’t work.

The demo is after the deadline.

**Lab Book Diary**

Note: The lab book must be photos from actual paper ( not note taking devices e.g. Microsoft surface etc) and pen( or pencil ). You can use coloured markers as well.

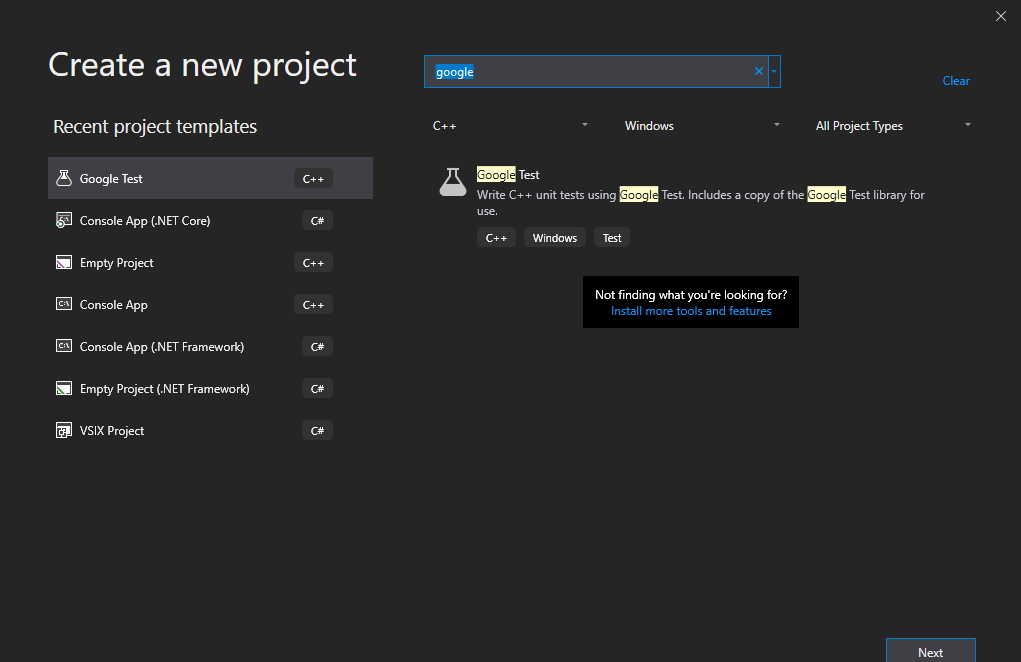
You must also show your lab book on screen when recording the demo video.

The lab book shows your work as you do it and illustrates your approach to solving the problems below. The lab book is worth 25%. The lab book must by written on paper and then take a photo of it and add it to a word doc or pdf. The lab book is not pseudo code!! It should show how you came up with your approach to the problems below, for instance drawing rows of boxes for arrays etc. It should be your rough work.

When you are demoing the lab, you will be asked questions about the code.

Upload the code to blackboard ,with the lab book as a pdf or word doc and a link to the video. You can upload the video to youtube or some other video sharing site and then send a link.

Create a googletest project in Visual Studio.



Copy and paste the contents of test.cpp ( from blackboard), into the default test.cpp in the googletest project.

Upload test.cpp to blackboard and your lab book word doc.

* Write a function to find an element in an array and prints out the index where it was found and -1 if it was not found.
* Write a function to find and return the second largest element in an array of positive integer, returns -1 if it can’t find one.
* Write a function to copy all elements from an array to another array. Both arrays are the same size.
* Write a function to insert an element in an array at a specified position. The function should insert an element at a particular index in the array and shift the elements further in the array to the right. The function uses a variable called count to track the number of elements in the array. The count cannot exceed the size of the array. The function should return true if an element was inserted, otherwise return false.
* Write a function to delete an element from an array of size 12 at specified position. The elements higher in the array are shifted to the left one position. The function uses a variable called count to track the number of elements in the array. The count cannot exceed the size of the array. The function should return true if an element was deleted, otherwise return false.
* Write a function to count the frequency of an element in an array.
* Write a function to count total number of duplicate elements in an array.
* Write a function to find the reverse of an array. You must only declare and use 1 array.
* Write a function to left rotate an array by one position. Note: the first element is rotated to the end of the array.
* You've built an inflight entertainment system with on-demand movie streaming. Users on longer flights like to start a second movie right when their first one ends, but they complain that the plane usually lands before they can see the ending. So you're building a feature for choosing two movies whose total runtimes will equal the exact flight length.
* Write a function that takes an integer flightLength (in minutes) and an array of integers movieLengths (in minutes) and returns a boolean indicating whether there are two numbers in movieLengths whose sum equals flightLength.
* When building your function:
* Assume your users will watch exactly two movies
* Don't make your users watch the same movie twice
* Write a function which will take as input an array of chars, and return the number of separate words, where a word is one or more characters separated by spaces. Your program should only count as words groups of characters in the ranges A..Z and a..z.
* For example the characters “ dd 3 f4 fff ff2 dd ” would return a word count of 3.