Lab 5 C++

Draft

Deadline February 8th

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

Create appropriate classes to represent questions and results.

You may use vector and string class in your code.

To Upload:

Code (lab5.cpp) All 3 parts in 1 file.

Lab Book (pdf or doc)

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

Use onedrive to store the video.

You must upload the code to blackboard before even if not all the questions are completed.

Upload code , video link and lab book before the deadline.

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

Lab Book Diary

You must also show your lab book, when demoing the code.

The lab book shows your work as you do it and illustrates your approach to solving the problems below.

You must create a  **5 to 10 minute video** explaining the code and how you came about the solution or how you tried to come up with a solution.

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 solution.

Contents of Video

Running the code from visual Studio.

Showing the tests you used to check your code.

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.

Part 1-40%

Write a multi choice quiz program .

The program will run and the user will be shown each multi choice questions in turn.

The order of the questions are randomised for every quiz.

Use the srand and rand() function to get a random number.

#include <time.h>

#include <stdlib.h>

using namespace std;

int main()

{

srand(time(0)); // seed the random number generator, otherwise the rand() will return the same sequence of numbers

int num = rand() % 10; // returns random int between 0 and 9.

}

The user inputs an answer using the keyboard, then “Correct” or “Wrong” is displayed and the user press a key to move onto the next question.

When all questions are completed the user is shown their result.

The user can repeat the quiz or exit.

Store the quiz data (questions, options and correct option) in a single file. The first number in the file should be the number of questions that are stored in the file.

Part 2. High Score table 20%

Add a high score table to the quiz, which is saved as a separate file. Before starting the quiz the user enters their name. When the user completes the quiz, their score is displayed and then the high score table of all users and their previous score is shown.

Part 3. Data stored as a json files 20%

Write a 2nd version of the quiz (in a separate project) which uses json to store the data. Use rapidjson library to read in the data in the format you create.

Part 4. SFML 20%

Add graphs for results in sfml of the players results. Use at least 2 chart types including a pie chart.