CS150 Spring 2016 Project TWO Game Time!

Due: **Electronic copy of .cpp file on Blackboard for lab no later than Tuesday March/2/2016 at 11:59 p.m.**
Late submissions will not be accepted under any circumstances.

Important:

Do NOT collaborate on this project.

All projects are to be completed individually.

Do NOT collaborate on this project.

Do NOT discuss the project with anyone other than the instructor and Teaching Assistants.

Do NOT collaborate on this project.

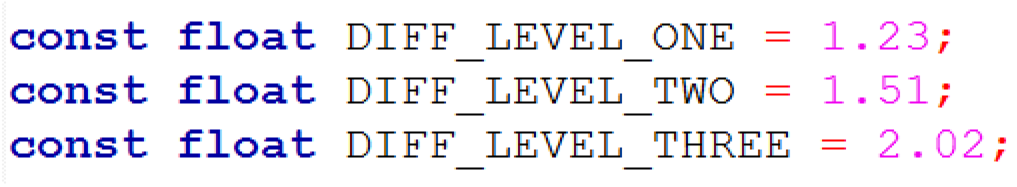
All projects are to be completed individually; duplicate programs will not receive credit.

Do NOT collaborate on this project.

Your code will be executed on the systems in Dragas. Make sure that your code compiles and runs on the systems in Dragas before you submit your .cpp file. Review the grading algorithm before you submit your project!

You have been tasked with writing part of a simple game program that will help beginner programmers study and understand C++.

The program will begin by prompting for the user information (the date and user game alias). The users' game alias is used as the name of the user's game score history output file. The format of this output file is the same as the sample input files provided for project one. Next the user is asked for the number of questions and the difficulty level of the game. If the number of questions entered by the user is not within the specified acceptable range (1-15), then repeatedly prompt the user until a valid value is entered. If the difficulty level entered by the user is not within the acceptable range (1-3), display the error message (shown in sample output below) and exit the program.
If user input is valid, display the questions and display the ouput as shown in the samples shown below.
A running score should be calculated, displayed at the end of the game, and written to the ouput file. Point values for the three difficulty levels are defined via three constant global variables as demonstrated in the sample below:



```
const float DIFF_LEVEL_ONE = 1.23;
const float DIFF_LEVEL_TWO = 1.51;
const float DIFF_LEVEL_THREE = 2.02;
```

The file is processed, and calculated results are displayed and also written to the default output file as shown in the sample output shown below.
You have to write an algorithm to solve this project. You are required to demonstrate the use of an algorithm in completing this assignment – meaning that you must include the number and short description of each step in the program documentation (comments)… showing your stepwise approach to solving this problem. See tips at bottom. Example output is shown at the end of this document.

Prompt the user to input the following *player information*, and echo the information to the screen.

- **Date** and Player **Alias** – strings that do not contain blanks
- **N**umber of Questions – an integer value (1 thru 15)
- **D**ifficulty Level – an integer value (1, 2, or 3)

Present the user with the desired number of questions, keeping a running score. Three input files are supplied with questions, and each contain questions of increasing difficulty as described below.

- **Open** the appropriate file and process the desired number of questions for each game attempt.
  - questions1.txt - difficulty level ONE
  - questions2.txt - difficulty level TWO
  - questions3.txt - difficulty level THREE

The name of the output file will be the same as the user alias, and will contain the *game score history:*

- **Open** the file and append the results for each game attempt.
  - **Date** – string that does not contain blanks (ex: Feb/14/2016)
  - **Level** – an integer value representing difficulty (1, 2, or 3)
  - **Score** of game – a floating point value.
  - See example output below.

Notes:
1. Remember to include programmer documentation in the source code and in the output.
2. Graders will follow the directions in your prompts. You should test your program with different data sets. Create your own question input files to test your solution.
3. Use a separate prompt for each user input value.
4. Format floating point values in the output with **3** decimal places.

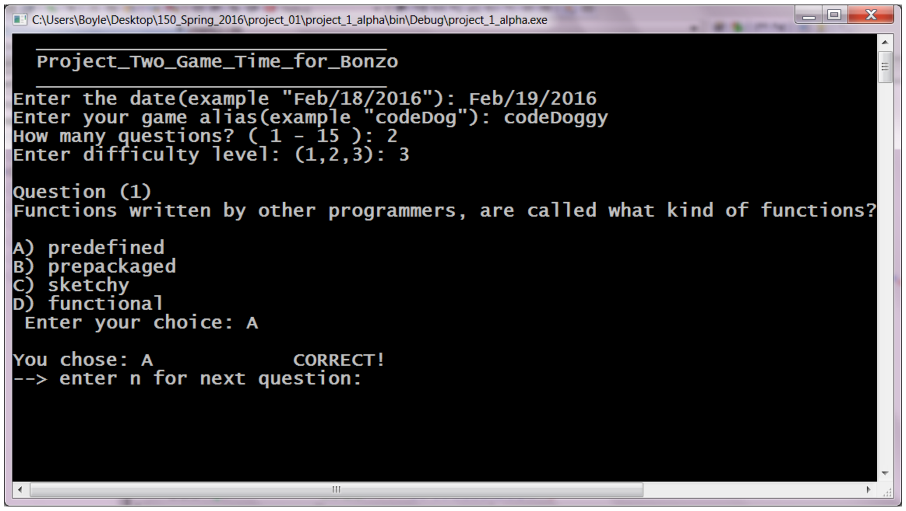
5. Remember to save a copy of the file you hand in. Leave the file unchanged after your submission until you receive a grade on the project assignment. Note that if you submit the wrong file as your solution, this will not earn you credit. You must submit your source code solution in the form of a file with the .cpp extension. Once the solution for a project has been released (usually coinciding with the due date), then no credit can be given or late submissions accepted. It is your responsibility to start early, and submit your solution in plenty of time to account for errors or questions for the lab TA on your submission.
6. Name your source code file using your last name and first initial as follows: lastname_firstInitial_Prj2.cpp ;for example, Meg Griffin would save her source code in a file name **Griffin_M_Prj2.cpp**
7. Remember that you can work in the Problem Solving Lab in Dragas 1103G or the lab located in room 3104 in the E & C S building. Hours for the labs are posted on the CS home page. You can also use the Remote Desk-Top Connection.
8. Do not discuss the project with anyone other than the instructor and Teaching Assistants. Do not collaborate on this project. Projects are to be completed individually; duplicate programs will not receive credit, and will be reported as a violation of the ODU Honor Code.

Submission details: Your TA will review the process of submitting your project in class. You will hand in an electronic copy of your source code. (.cpp file) We will run your program so it is important that you hand in a copy of your source file (**Griffin_M_Prj2.cpp). The file you hand in must have a .cpp extension.**
You need to start this project early – do NOT wait until the last day to complete and submit your solution. Consult with your TA BEFORE submitting, if you have any questions. **Your code will be executed on the systems in Dragas**. ***Your code must compile and run on the systems in Dragas.***

Sample Input and Output

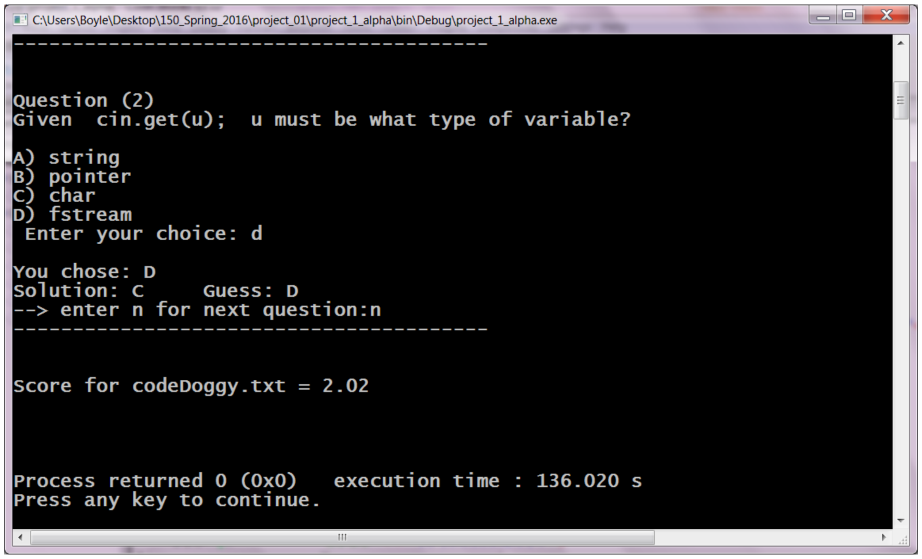
Output to monitor

Output to file



```
C:\Users\Boyle\Desktop\150_Spring_2016\project_01\project_1_alpha\bin\Debug\project_1_alpha.exe

  Project_Two_Game_Time_for_Bonzo

Enter the date(example "Feb/18/2016"): Feb/19/2016
Enter your game alias(example "codeDog"): codeDoggy
How many questions? ( 1 - 15 ): 2
Enter difficulty level: (1,2,3): 3

Question (1)
Functions written by other programmers, are called what kind of functions?

A) predefined
B) prepackaged
C) sketchy
D) functional
 Enter your choice: A

You chose: A           CORRECT!
--> enter n for next question:
```

	difficulty	raw
date	level	score
Nov/1/2015	1	3.996
Nov/1/2015	1	15.111
Dec/8/2015	1	21.919
Jan/6/2016	2	91.014
Feb/1/2016	2	4.445
Feb/3/2016	2	81.001



```
C:\Users\Boyle\Desktop\150_Spring_2016\project_01\project_1_alpha\bin\Debug\project_1_alpha.exe
----------------------------------------


Question (2)
Given  cin.get(u);  u must be what type of variable?

A) string
B) pointer
C) char
D) fstream
 Enter your choice: d

You chose: D
Solution: C     Guess: D
--> enter n for next question:n
----------------------------------------


Score for codeDoggy.txt = 2.02



Process returned 0 (0x0)   execution time : 136.020 s
Press any key to continue.
```

Three sample input files are provided: questions1.txt **and** questions2.txt, **and** questions3.txt

Helpful Tips

- Start Now. Do not procrastinate.
- Examine the sample code from our textbook, and run the programs in your IDE (codeblocks).
- Go to the recitations to get help from the TAs.
- Go see the TAs during their office hours.
- The algorithm steps you write will prove useful in creating and implementing your own solution.