CSCI2120 Introduction to Software Engineering
Assignment 1
Deadline: 28 Feb 2016 (Sun) 23:59

Objective

The objective of this assignment is to practice Object-Oriented Programming (OOP) in C++.

Task

In this assignment, your task is to write a game that allows a human player to guess a 4-digit password in six trials.

Before the Game Starts

Before the game starts, your program has to ask the player to input his/her name. The player name has to contain at least one alphabet character (i.e., 'A' – 'Z' or 'a' – 'z'). If not, your program will ask the player to enter the name again until the player name is valid. In addition, the name may contain space characters. (Hint: use **`getline(cin, nameOfPlayer)`** to read the input, where **`nameOfPlayer`** is a string variable. You have to type **`#include<string>`** in the beginning of the code. After using **`getline(…)`**, please avoid using "**`cin >>`**" in your program as they are incompatible with each other.)

Game Flow

When the game starts, your program will randomly generate a 4-digit password with no duplicate digits. For example, 1740 is a valid password, but 1741 is not. The password will be guessed by the player in at most six trials. The message "**`<player_name>, you have to guess a 4-digit password in 6 trials!`**" is shown.

In each trial, your program should display "**`Make a guess:`**" to ask the player to input a 4-digit number with no duplicate digits. If the input format is not correct (e.g., consists of more than or less than 4 digits, contains duplicate digits), your program should display the message "**`Your guess should be a 4-digit number with no duplicate digits.`**" and ask the player to input again until the input format is correct (this will consider the same trial). After receiving a guess in correct format, your program will check the guess of the player against the password generated when the game starts. If the guess is not correct, four hint symbols will be displayed to give a hint to the player and the number of trials left also will be displayed, where the message is "**`<four_symbols> (Number of trials left: <number_of_trials_left>)`**". Table 1 shows the symbols and their semantics.

Hint Symbol	Semantics
O	Correct number in correct position
.	Correct number but in wrong position
X	Wrong number

Table 1: Different hint symbols and their semantics.

For example, if the password is 1740 and the player's guess is 1234, then the four hint symbols will be "OXX.". If the player can guess the password in six trials, the game ends and the message "**<player_name>, you have cracked the password! Congratulations!**" will be displayed. If the player uses up all the trials, the game ends and the message "**<player_name>, you have used up all trials! The password is <password>! Game Over!**" will be displayed. For both cases, the program ends after displaying the aforesaid message.

Additional Requirement

You are required to implement this assignment in OOP style with the principles of encapsulation and information hiding. Your program should contain at least three classes, namely **PasswordPanel**, **Player**, and **PasswordGuessingGame,** and your main function should not contain more than 30 lines of code.

Assume that the password is 1643. Two sample runs are shown as follows (underlined bold characters are input from the user):

```
Please enter your name: Agent 13
Agent 13, you have to crack a 4-digit password in 6 trials!
Make a guess:
0123
X.XO (Number of trials left: 5)

Make a guess:
1453
O.XO (Number of trials left: 4)

Make a guess:
1643
Agent 13, you have cracked the password!  Congratulations!
```

```
Please enter your name: 123
Your input has to contain at least one alphabet character.
Please enter your name: 12 3
Your input has to contain at least one alphabet character.
Please enter your name: Me
Me, you have to crack a 4-digit password in 6 trials!
Make a guess:
0120
Your guess should be a 4-digit number with no duplicate digits.

Make a guess:
a123
Your guess should be a 4-digit number with no duplicate digits.

Make a guess:
a1234
Your guess should be a 4-digit number with no duplicate digits.
```

```

Make a guess:
0123
X.XO (Number of trials left: 5)

Make a guess:
3210
.X.X (Number of trials left: 4)

Make a guess:
4567
.X.X (Number of trials left: 3)

Make a guess:
5678
XOXX (Number of trials left: 2)

Make a guess:
7890
XXXX (Number of trials left: 1)

Make a guess:
0987
XXXX (Number of trials left: 0)

Me, you have used up all trials!  The password is 1643!  Game Over!
```

Declaration

As required by the University, please place the following declaration as the comment in the beginning of your source code.

```
' CSCI2120 Introduction to Software Engineering
'
' Assignment <assignment_number>
'
' I declare that the assignment here submitted is original except for source
' material explicitly acknowledged.  I also acknowledge that I am aware of
' University policy and regulations on honesty in academic work, and of the
' disciplinary guidelines and procedures applicable to breaches of such policy
' and regulations, as contained in the following University website:
' http://www.cuhk.edu.hk/policy/academichonesty/.
'
' Student Name: <your_name>
' Student ID: <your_student_ID>
' Date: <date>
```

Assessment Scheme

Correctness: 70%

Class Design: 20%

Program readability (e.g., indentation, comments, meaningful variable names, etc.): 10%

Please make sure that your program can compile successfully or you will get a very low score. Please also be reminded that your TA will read your code and can identify similar submissions easily. Consequences of confirmed plagiarism cases can be failing of the course and receiving a demerit.

Testing Platform

Your program will be tested in the following platform:

- Windows 7
- Visual Studio 2013

Submission

Please follow the following steps to submit your work.

Step 1: If you only have one .cpp source file, please save your source file as <your_student_id>*_assg1.cpp*. (e.g. 1155012345_assg1.cpp). If you have more than one source files, please put them into a compressed file (zip, rar, or 7z) and name the compressed file as <your_student_id>*_assg1.zip*. (or .rar or .7z) (e.g. 1155012345_assg1.zip)

Step 2: Go to CU eLearning (http://elearn.cuhk.edu.hk/) and login.

Step 3: Go to "Assignments → Assignment 1 → Assignment Submission"

Step 4: Upload your file prepared in Step 1.

Resubmissions are allowed. But only the latest one will be counted. Late submissions within three days can only receive 70% of the marks. Late submissions more than three days will not be graded.