cisc1110, fall 2010, assignment for unit IV, part 2

This is the assignment for unit IV, "input; variables." You are expected to complete the assignment in the C++ language and submit your ".cpp" file. You must complete and submit the assignment on or before the due date of **October 12**. *Remember: Assignments will NOT be accepted more than 7 days late.*
This assignment is worth **5 points**.

Submission instructions:

- Submit your assignment to me via email: `sklar@sci.brooklyn.cuny.edu`
- Your email subject line should be: **CISC 1110 Lab IV submission**
- Attach your C++ (**roulette.cpp**) file to your email.
- Make sure your name is in the body of the email message.
- Make sure your name is also included in the header comments at the top of your C++ file.

Overview

For this assignment, you will create a program that simulates a modified version of the game of roulette. If you do not know what roulette is, then you can read about it here: `http://www.classicasinos.com/Roulette`. But don't worry about the real rules, because our game is somewhat modified.

Our roulette wheel has 11 spaces: numbered from 0 to 10. Your program will ask the user to place a bet on one of those numbers. Then the computer will "spin" the wheel and compute how much money the user has won. The user will have 3 rounds to play the game.

Follow the step-by-step instructions below to create our version of this game. There are **6** steps, described below, for you to follow in order to develop your program. Make sure that for each step, you compile, run and test your program to make sure it works as described in the instructions for that step. Feel free to add extra `cout` statements to print intermediate values to help you debug your program.

You should make copies of the program along the way, after each step, so that you save the progress you make in your work—in case of disaster or you need to backtrack to a previous step that worked.

Your program should have a *header comment* at the top of the file, which should include your full name, the date you created the program and the filename. It can also include useful comments describing what the program does.

step 1 (0.7 points)

- Create a new program file named **roulette.cpp**.
- Initialize the random number generator.
- Declare two arrays of 11 integers. One will represent the roulette wheel (call it `wheel`), and one will represent the player's bets (call it `player`).
- Initialize all 11 entries in both arrays to 0.
- Compile, run and test your program. Go back and fix it if it doesn't work properly.
- *HINT:* You might want to print out both arrays to make sure that you have initialized them correctly.

step 2 (1 point)

- Modify your program as follows.
- Declare an integer variable that will store the amount of money the user has in her pocket. Initialize this variable to 1000.
- The rest of this step, along with steps 3, 4 and 5, comprise one round of the game. Since the user is given 3 rounds of play, you should use a `for` loop to iterate 3 times. The body of the loop will contain the code for one round. This means that the code you will write for the remainder of this step, plus the code for steps 3, 4, and 5, will all be inside the body of the `for` loop.
- Prompt the user to enter a number between 0 and (including) 10. Read in the user's number and store it in an integer variable (call it `guess`).
- Prompt the user to enter the amount of money they want to bet. Read in the amount of their bet and store it in an integer variable (call it `amount`).
- Display a message echoing back the user's bet.
- In the `player` array, set the element at the index of the user's guess to the amount the user entered; e.g.:
  ```
  player[guess] = amount;
  ```
- Compile, run and test your program. Go back and fix it if it doesn't work properly.

step 3 (1 point)

- Modify your program as follows.
- Declare an integer variable to represent the computer's "spin" (call it `spin`), and set that variable to a random number between 0 and (including) 10.
- Display a message telling the user what number the computer spun.
- In the `wheel` array, set the element at index of the computer's spin to 1; e.g.:
  ```
  wheel[spin] = 1;
  ```
- Compile, run and test your program. Go back and fix it if it doesn't work properly.

step 4 (1 point)

- Modify your program as follows.
- Now compute how much the user won on that round. Do this using a `for` loop (which will be *nested* inside the `for` loop you started in step 2). First, initialize an integer subtotal to 0. Next, write a `for` loop that counts from 0 up through (and including) 10. In the body of the loop, multiply each element of the `wheel` array by the element in the `player` array stored at the same index, and add that to your subtotal; e.g.:
  ```
  subtotal = subtotal + ( wheel[i] * player[i] );
  ```
  After the loop is finished, the subtotal will contain the amount of money that the user has won, if anything. Note that the subtotal will be 0 if the user did not guess the same number that the computer "spun".
- Display a message telling the user how much she won in that round and how much money she has left in her pocket.
- Compile, run and test your program. Go back and fix it if it doesn't work properly.

step 5 (1 point)

- Modify your program as follows.
- Zero out the `wheel` and `player` arrays to get ready for the next round. You can do this by just resetting the entries you set in steps 2 and 3 to 0.
- Loop back to the beginning of a round, as described in step 2.
- Compile, run and test your program. Go back and fix it if it doesn't work properly.

step 6 (0.3 points)

- Modify your program as follows.
- After the user has played 3 rounds (i.e., after your `for` loop that iterates through the rounds), display a "GAME OVER" message and display how much money the user has left in her pocket.
- Compile, run and test your program. Go back and fix it if it doesn't work properly.

sample output

Below is sample output from my solution to the assignment:

```
Welcome to my Roulette Game!
You will have 3 turns to play, and you start with $1000 in your pocket.
On each turn, you will be asked to enter a number between 0 and 10,
and an amount to bet that your number will come up on the wheel.
Then the computer will spin the wheel...and we'll see how much you win (or lose!)

This is turn #1 of 3 and you have $1000 in your pocket.
Enter your bet (i.e., enter a number between 0 and 10): 5
How much do you want to bet that your number will come up? 5
You have bet $5 on number: 5
The computer spun number: 9. You have won $0 on this round.

This is turn #2 of 3 and you have $995 in your pocket.
Enter your bet (i.e., enter a number between 0 and 10): 5
How much do you want to bet that your number will come up? 5
You have bet $5 on number: 5
The computer spun number: 5. You have won $5 on this round.

This is turn #3 of 3 and you have $1000 in your pocket.
Enter your bet (i.e., enter a number between 0 and 10): 5
How much do you want to bet that your number will come up? 5
You have bet $5 on number: 5
The computer spun number: 7. You have won $0 on this round.

GAME OVER!
You have $995 in your pocket.
```

did you notice?

...that the program is not robust. It does not check to see if the user enters valid numbers between 0 and 10, nor does it check to see if the user tries to bet more money than she has in her pocket. You will learn how to do things like this in the next unit!