For this assignment you will write a C# program which will play the game of *Flaunt*. Flaunt takes two players who take turns guessing numbers from 1 to 5. At each turn each player gets the number that she guesses added to her score *unless* her number is exactly one less than her opponent's. If you guess exactly one less than your opponent then you get both numbers added to your score and your opponent gets nothing. For example, if I guess 3 and you guess 5 then I get 3 points and you get 5 for that turn. On the other hand if I guess 3 and you guess 4 then I get 7 points and you get none. There is one more complication in the rules. If one player uses the number k then she gets $k^n$ points where $n$ is number of successive turns in which k is repeated. For example, if I guess 4 and you guess 2, I get 4 points and you get just 2. If I guess 4 again and you guess 1 then I get 16 points and you get just 1. If, on the next turn I guess 4 again and you guess 3 then you get $3 + 4 \times 4 \times 4 = 67$ points added to your score and I get nothing.

For this assignment the computer *must* be one of the players. It will guess one of the numbers and it will keep track of and display the score after each round. Flaunt continues until one player scores at least 60 points. The person (or computer) who scores at least 60 points first is declared the winner.

Obviously, the way to win this game is to know in advance what number your opponent is going to guess. Since mind reading is not a C# function we will use random guesses for the computer's choice of a number. C# has a random number generator as a built in function. Pages 245-246 in your Deitel & Deitel or pages 401-402 in Nakov and Kolev explains how to use this function.

Your program should declare a winner and allow for ties (both players exceed 60 points in one turn). Set up the input as a function so that it will not accept anything from the user except a number from 1 to 5. To do this you will need a loop that continually does the input and generates error messages until the user enters a valid number.

Your program *must* be modular - that is, it should consist of a main program that is largely a sequence of function calls.

A typical output screen for a partial game of *flaunt* is shown below. Bold text shows user input.

```
Welcome to the game of flaunt!
Guess a number 1 to 5... 4
The computer guessed 3.
The score is Computer 7  Player 0

Guess a number 1 to 5... 5
The computer guessed 2.
The score is Computer 9  Player 5
```
... 
Guess a number 1 to 5... 3
The computer guessed 4.
The score is Computer 54  Player 62
The Player wins.

Another game?? (Y/N) N

Turn in a complete design for your program and your zipped project folder to
\\cecsfp01\Users\Everyone\Engr123