Lab 15: do and for loops **Due:** 11/9/22

In this lab you will practice using do-while and for loops to perform repetitive tasks. In this lab you will review and write two functions that each demonstrate the different types of loops.

Function main():

Just calls the two functions described below. For each of them, it:

1) Calls the function.
2) Pauses the program to show the results.
3) Clears the screen (except after the last call).

Function guessNumber():

The **carInsComplex-do_while.cpp** program provided illustrates the skills you are learning in this lab. Open it in your IDE, read and understand the code and finally run the program with different inputs to see how it works.

The **xpowy-for.cpp** program provided illustrates the skills you are learning in this lab. Open it in your IDE, read and understand the code and finally run the program with different inputs to see how it works.

Your function

For your function you must make some improvements to the Guess Number program created before.

To allow the user to decide whether to use a fixed seed (to easily test the program while developing it) or a seed based on the computer's time (to generate a different set of random numbers every time you run your program) you have to use a do-while loop to ensure the user enters a whole number within the range 1000..4000 (to use it as the fixed seed) or a zero (to get the seed from the computer's time). If the user enters a value that does not fit this criterion your program must keep asking for the value until a valid one is entered.

To allow the user up to 5 guesses you must use a for loop. If their guess is correct use a break statement to exit the for loop.

To keep your main() function simple:

1) You **must** define function **randNumGen()** to generate a random number. This function receives two whole numbers: the first one is the upper boundary and the second one is the lower boundary used to generate a random number. It returns the random number generated using expression:

(rand() % (highRange - lowRange + 1)) + lowRange

2) You **must** define function **getSeed()** to get the number for the seed from the user. This function does not receive anything from the caller. It uses a **do-while loop** to get a seed from the user. The seed must be a zero or a number within the range 1000..4000. If the user enters an invalid value, the function must keep asking until a valid value is entered. Once a valid value is entered it must be returned to the caller as a whole number.

3) You **must** define function **guessNumber()** to implement the guessing part of the game. This function receives a number and uses a **for loop** to give the user five chances to guess it. As the user provides a guess the program indicates whether the guess was correct or if it was too low or too high to help her/him. If the user runs out of guesses it shows her/him the number.

Function `pigRoll()`:

A condition-controlled while loop will repeat as long as the given condition is satisfied.
Use the break statement to immediately exit a loop.

Example Program: Condition-controlled while Loops

The **carInsComplex-while.cpp** program provided illustrates the skills you are learning in this lab. It allows the user to calculate insurance premiums until the user chooses to quit. Open it in your IDE, read and understand the code and finally run the program with different inputs to see how it works.

Example Program: break

The example program for break is a modification of the carInsComplex-while.cpp program. Instead of using a condition-controlled while loop, this program uses an infinite loop. When the user decides to quit, the program uses break to exit the infinite loop. The **carInsComplex-break.cpp** program provided illustrates the use of break in loops. Open it in your IDE, read and understand the code and finally run the program with different inputs to see how it works.

Your function

In your function you will implement part of the Pig game using loops and the expressions learned in previous labs to generate random numbers. For this lab you will begin working on the computer player for the Pig game. You will write the code for the computer to take one turn at rolling the die. During a turn in the Pig game, the player can roll the die as many times as he wants as long as he does not roll a 1 (in which case he loses his turn without scoring any points).

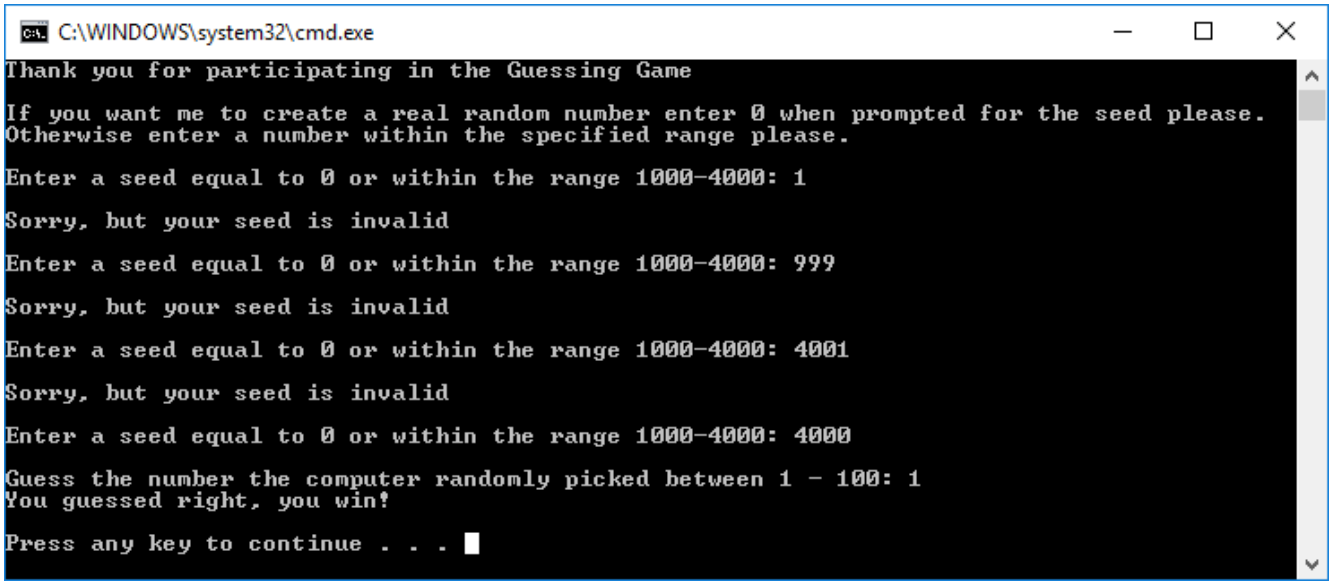
In order to create a computer program that simulates a Pig player we need to develop a playing strategy that the program will implement. We will have our computer player use a simple strategy: it will roll until its turn total is 20 or greater, or until a 1 is rolled.

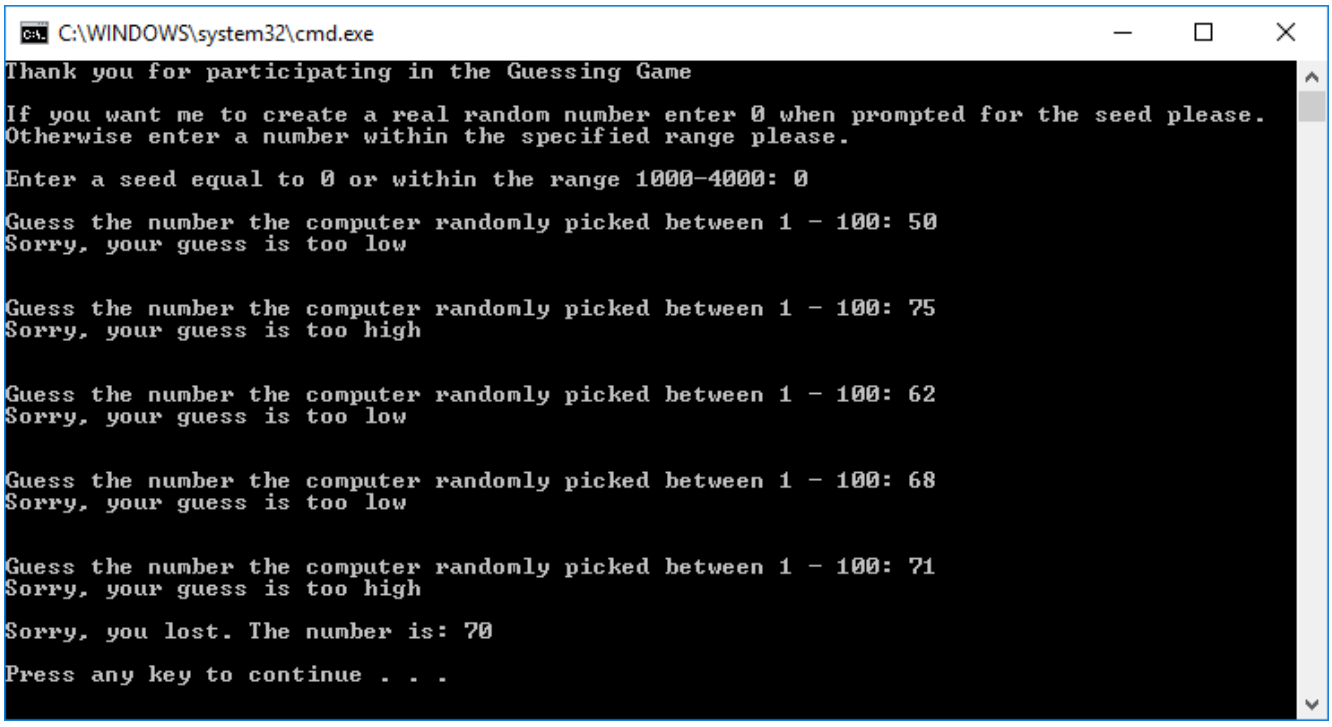
Open **lab15.cpp** in your IDE and implement the algorithms provided in the source code as comments.

I am posting my solution for your reference. Please run your program as many times as necessary to ensure that you tested it thoroughly. Make sure it behaves like my sample solution.

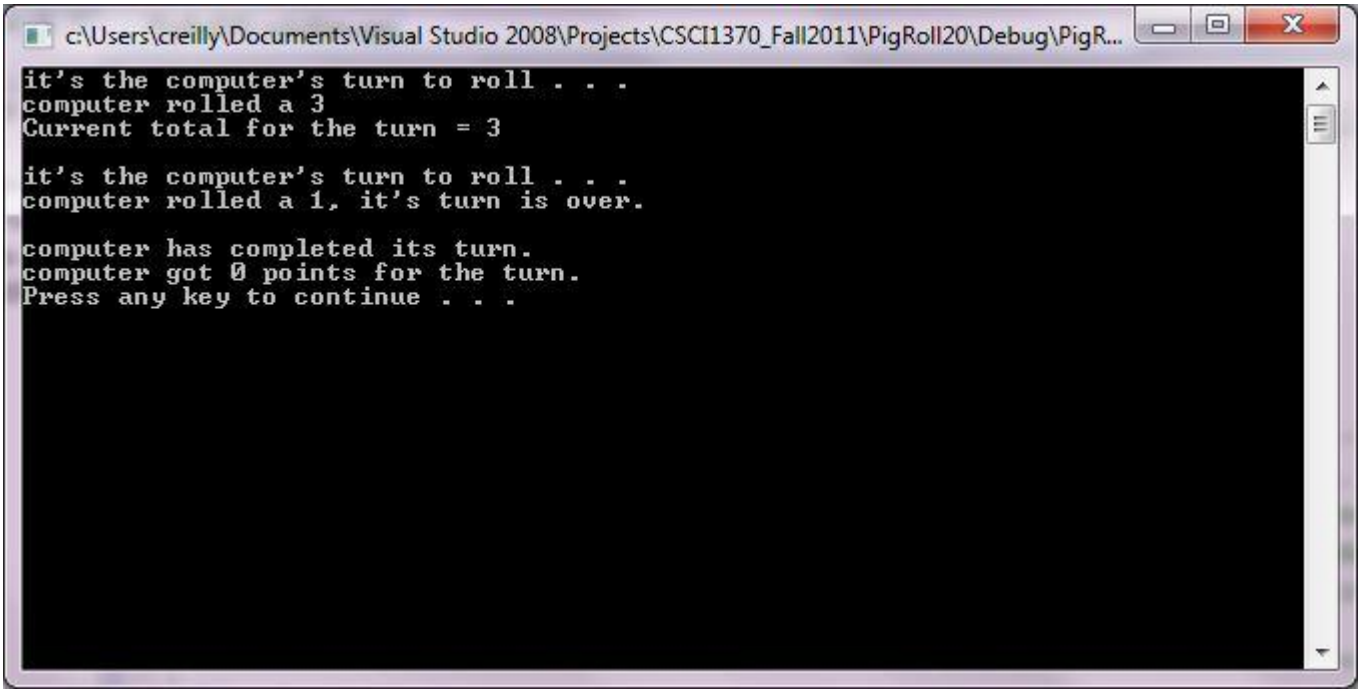
https://replit.com/@GDietrich/1370-lab15sample

Sample run of function guessNumber





Sample run of function pigRoll



If you have concerns or specific questions, post them on the Discussion Board of Blackboard.
Don't forget to include at the top of the program the comments shown below with your information (name, class and section number, etc.)

```
////////////////////////////////////////////////////////////////////
//
// Name: <Put your name here>
// Date: <Today's date>
// Class: <Your class number and section number, like: CSCI 1370.02>
// Semester: <This semester, like: Spring 2012>
// CSCI 1370 Instructor: <Your lecture instructor's name>
//
// Program Description: Enter here your description of what the program does
//
////////////////////////////////////////////////////////////////////
```

When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.

Paste the link to your solution and the source code in the textbox corresponding to Text Submission (click on the Write Submission button) before you click on Submit.

The following is the basic criteria to be used to grade your submission:

You start with 100 points and then lose points as you don't do something that is required.

-5: Wrong variable types
-5: Missing comments at the top of the program
-5: No comments or too few comments in source code
-20: Missing implementation of function **randnumgen()**
-25: Missing implementation of function **getSeed()**
-30: Missing implementation of function **guessNumber()**
-25: Doesn’t use a do-while loop in getSeed()
-10: Incorrect implementation of the do-while loop in getSeed()
-25: Doesn’t use a for loop in guessNumber()
-10: Incorrect implementation of the for loop in guessNumber()
-10: Doesn’t loop 5 times if no correct guess
-10: Doesn’t exit loop when guess is correct or after guessing five times
-10: Doesn’t print win/lose message
-30: Missing implementation of function **pigRoll()**
-10: Incorrect implementation of the while loop in pigRoll()
-10: Doesn't stop rolling when the total is 20 or greater in pigRoll()
-10: Doesn't stop rolling if a 1 is rolled in pigRoll()
-5: Doesn't calculate correct total of rolls (other than 1) in pigRoll()
-5: Doesn't set turn total to 0 when a 1 is rolled in pigRoll()
-5: Incorrect input/output format
-50: Program doesn't compile
-40: Program doesn't run to completion
-20: Program does not implement the provided algorithm
-20: Incorrect/missing source code or incorrect/missing link to your Repl.it solution
-100: The code submitted is not your creation (you got it from a web site or another person)
-10: Late

Note: more points may be lost for reasons not specified here.