Homework 5 CSS 142

Instructions: Please submit your files to the Canvas submission page for this assignment by the due date. Late homework will not be graded.

Grading Rubric:
70% for correctness - includes: following the requirements of each problem as stated (it is your responsibility to ask for clarifications); using the concepts learned from class; testing your code for correctness; having code that works.
30% for style - includes: commenting (block and in-line); spacing between operator and operands; obeys style guidelines indicated during lectures.

Problem 1. (*Guessing Game*) In this problem you will be creating a game that you play against the computer. Make a class named GuessingGame - when executed, this program should randomly generate a secret integer from 1 to 100, and then instruct the user to try to guess what it is. After each guess the program will tell them if they are too high, too low, or if they got it right - and if they got it right it will congratulate them, tell them how many guesses it took them, and offer to either play again or exit the program.
An example interaction with this program follows:

```
---------------------------------------Please guess a number
between 1 and 100:
50

Too low! Guess again: 75

Too high! Guess again: 63

Good job, you got it after 3 guesses! Want to play again (Y or N)?
N

Thanks for playing!
----------------------------------------
```

After you have made this game, play it some and try to figure out the best strategy. In a text file called HowToGuess.txt, explain the best strategy you employed to play the game and about how many guesses it usually took you to get the answer.

For this problem, you should submit to Canvas two files: GuessingGame.java and HowToGuess.txt.

Problem 2. (*Compound Interest*) If an amount a is invested at an interest rate r compounded n times per year, then the final value f at the end of one year is given by $f = a(1 + r/n)^n$. This is the familiar formula for compound interest (so if we compound annually, then n = 1). Typically, compounding is done quarterly (n = 4), daily (n = 365) or continuously ($n \rightarrow \infty$).

In a class called CompoundInterest, write a method that implements the compound interest formula f above. This method should take the following parameters: a double a (as in the given formula), a double r (as in the given formula), and an int n (as in the given formula). This method should return the value of the investment at the end of a year (as specified by the formula f). You should NOT use any Math libray functions or any imports (*hint: you will need a loop*).

Test your method in main using an initial investment of $100.00, an interest rate of 5 percent (i.e. r = 0.05), and the following values for the compound number: 1, 4, 365, 1,000, and 10,000.

Explain your results in your program's comments. For this problem, you should submit to Canvas one file: CompoundInterest.java.

Problem 3. (*File I/O*) The Fibonacci numbers F_n are defined as follows: F_0 is 0, F_1 is 1, and

$$F_{i+2} = F_i + F_{i+1} \text{ for } i = 0,1,2,...$$

In other words, each number is the sum of the previous two numbers. The first few Fibonacci numbers are 0, 1, 1, 2, 3, 5, and 8.

One place where these numbers occur is as certain population growth rates. If a population has no deaths, then the series shows the size of the population after each time period. It takes an organism two time periods to mature to reproducing age, and then the organism reproduces once every time period. The formula applies most straightforwardly to asexual reproduction at a rate of one offspring per time period. In any event, the green crud population grows at this rate and has a time period of five days. Hence, if a green crud population starts out as 10 pounds of crud, then in 5 days, there is still 10 pounds of crud; in 10 days, there is 20 pounds of crud; in 15 days, 30 pounds; in 20 days, 50 pounds; and so forth.

Write a program that takes both the initial size of a green crud population (in pounds) and a number of days as input from a file greencrud.txt (organized as initial size 'space' number of days), and outputs the number of pounds of each green crud population after that many days to an output file called crudout.txt (make sure each output is labeled somehow). Assume that the population size is the same for four days and then increases every fifth day.

Input file: greencrud.txt
For this problem, you should submit to Canvas one file: Greencrud.java.

Problem 4. (*Basic Debugging*) Consider the code below:

```
public class Average {
   public static void main(String[] args) { int n = 1.0;
      Scanner scanner = new Scanner(system.in);
      System.out.println("This program calculates " +
                     "the average (mean) of entered numbers."); int x; do
      {
         System.out.println("Enter another number " +
                      "(or 0 to stop and show the average): "); x +=
         scanner.next(); n++;
      } while (x != 0);
      System.out.println("Your final average is: %d" + x / n); }
}
```

Find and fix all the bugs in this program. Turn in to Canvas a file called Average.java containing the correct version of this code with in-line comments explaining each fix.