Assignment 2

Due Sunday Oct 12, 2014 at midnight

Read the instructions below carefully. The instructions must be followed. The assignment is worth 5% of your final grade. No late assignments will be accepted.

Main goal of this assignment is help you learn how to develop your own methods while also learning how to use if statements and loops.

For this assignment, you are allowed to use only concepts that you have seen in the first 4 weeks of class. For example, you can use if statements and loops (but not arrays).

Write a different java program for each question. Each of these programs should be placed in a separate file called A2Qx.java where x is replaced by the question you are answering. Follow the instructions you have received in the first lab on how to put all these files in a folder, zip that folder and submit it via Blackboard Learn.

The questions that do not compile will be graded with mark 0. For that reason, if you run out of time and/or one of your answers contains code that does not compile, then comment out that section of the code.

At the beginning of each program you should have:

```
/*
==================================================
question X
==================================================
*/
```
(Replace X with the number of the question you are answering.)

Questions:

1. (2 points) Triangle or not?

If you are given three sticks, you may or may not be able to make a triangle with them. For example, if one of the sticks is 10cm long and the other two are 1cm long, you will not be able to form a triangle. For any three lengths, there is a simple test to see if it is possible to form a triangle with them:
"If any of the three lengths is greater than the sum of the other two, then you cannot form a triangle. Otherwise, you can."
In mathematics, this fact is known as the "Triangle Inequality".
http://en.wikipedia.org/wiki/Triangle_inequality

Write a program that has a main method and a method called isTriangle. The method isTriangle takes three numbers as input parameters, and returns either true or false, depending on whether one can form a triangle from sticks with the given lengths.(To avoid numerical errors, you may assume that the lengths are given as integers).
The main method should prompt the user to input three positive integers (you may assume that the user types in a correct input). The main method should them call/invoke isTriangle method to determine if they can form a triangle. If the method returns true, print the message "The three lengths determine a triangle", otherwise, print the message as indicated in the examples.

Examples:

Your program: Enter three positive integers for lengths
User: 20 20 30
Your program: The three lengths determine a triangle.

Your program: Enter three positive integers for lengths
User: 2 30 1
Your program: There is no triangle whose sides have length 2, 30, and 1.

2. (2 points) Safe number?
Write a program that has a main method and a method called is_safe. The is_safe method tests if a number is safe to use during a game of "Yo-Jo". A number is not safe if it contains a 9 as a digit, or if it can be divided by 9. The main method should prompt the user to input a number between 1 and 99 (you may assume that the user types in a correct input) and calls is_safe method to test if the number is safe or not. If the method returns true, print the message "The number is safe", otherwise, print "The number is not safe".

Examples:

Your program: Enter a number between 1 and 99
User: 93
Your program: The number is not safe.

Your program: Enter a number between 1 and 99
User: 82
Your program: The number is safe.

Your program: Enter a number between 1 and 99
User: 29
Your program: The number is not safe.

Your program: Enter a number between 1 and 99
User: 36
Your program: The number is not safe.

3. (2 points) Ontario "Drive Clean" program
Assume that the Ontario "Drive Clean" program requires that a car must pass an emissions test for licence renewal if both of the following conditions are met:

- The car's model year is odd and the current year is even, or vice versa.
- The difference between the current year and the model year is at least 3 but no more than 20.

Write a program that has a main method and a method called test_needed. The test_needed method determines if a car needs a Drive Clean emission test. The main method should ask the user to input the model year of the car and the current year. It should then call the test_needed method with those values. The test_needed method returns true if an emission test is needed and false otherwise. Finally, the main method should print a message to say if a test is needed or not.

Examples:

Your program: Enter the model year
User: 2005
Your program: Enter the current year
User: 2014
Your program: An emission test is needed.

Your program: Enter the model year
User: 2005
Your program: Enter the current year
User: 2013
Your program: An emission test is not needed.

4. (2 points) Compute a series using a loop
Write a program that has a main method and a method called compute_series. The compute_series method computes the value of $1 + 1/2^2 + 1/3^2 + 1/4^2 + ... + 1/n^2$. The main method should prompt the user to input the value of n, and then it should call compute_series method by passing that value. Finally it should print the result returned by the compute_series method.

5. (2 points) Draw a rectangle using stars.
Write a program that has a main method and a method called draw_rectangle. The main method should ask the user to input the length of the rectangle (an integer greater than 1). Then the draw_rectangle method should be called, with one parameter, the length. The draw_rectangle method should print m stars (where m is the length); then on the next line it should print one star, m-2 spaces and one more star. Then on the third line it should print m stars.

Examples:

Your program: Enter an integer greater than 1 for the length
User: 7
Your program:
```
*******
*     *
*******
```

Your program: Enter an integer greater than 1 for the length
User: 20
Your program:
```
********************
*                  *
********************
```

6. (4 points) Draw a tree using stars.
Write a program that has a main method and a method called draw_tree. This method takes as input an integer, m. This integer determines the number of lines the tree must have (how tall the tree is). You may assume that m is bigger or equal to 1. The method should draw the crown of the tree (the upper part), as follows. On the first line, print m-1 spaces and one star; on the next line, print m-2 spaces and 3 stars; on the next line, print m-3 spaces and 5 stars, etc., until we have m lines. At the end, add the bottom part of the tree, made of exactly two lines regardless of the value of m, each of the 2

lines having m-1 spaces and one star. So the tree has m+2 lines in total, m for the upper part and 2 for the bottom part.
The main method should prompt the user to enter an integer (for the height of the tree) and then call the draw_tree method with that value.

Examples:

Your program: Enter an integer (how tall is your tree)
User: 5
Your program:
```
    *
   ***
  *****
 *******
*********
    *
    *
```

Your program: Enter an integer (how tall is your tree)
User: 8
Your program:
```
       *
      ***
     *****
    *******
   *********
  ***********
 *************
***************
       *
       *
```

Your program: Enter an integer (how tall is your tree)
User: 3
Your program:
```
  *
 ***
*****
  *
  *
```

7. (6 points) Guessing game.
In this question, you will develop a guessing game, in which the user will play against a computer. A number between 1 and 100 will be secretly chosen by the computer and the player will try to discover this number by repeatedly guessing numbers. If a guess is greater than the chosen number, the program prints out '>', if the guess is smaller than the chosen number, the program prints out '<', and otherwise the guess is correct and the player wins.

The number of guesses is limited. The maximum number of guesses is determined by the player who indicates how difficult the game should be. There are three levels of difficulty, including easy (between 20 and 30 guesses), difficult (between 10 and 15 guesses), and very difficult (between 1 and 5 guesses). The user determines the desired level by entering 1 for easy, 2 for difficult and 3

for very difficult. For example, if the user wishes an easy game he/she will enter 1 and the program generates a random number between 20 and 30. This number determines the number of guesses the user is allowed. The following shows an example output for a game.

Your program:
Do you want to play a
1. Easy game?
2. Difficult game?
3. Very difficult game?
Make a choice (1, 2, or 3):
User: 4
Your program:
Invalid Choice!
Make a choice (1, 2, or 3): 1
You have 22 guesses to find the secret number.
Guess 1 of 22:
User: 30
Your program: The guess is too big.
Guess 2 of 22:
User: 10
Your program: The guess is too small.
Guess 3 of 22:
User: 120
Your program:
The number 120 is not valid.
Guess 3 of 22:
User: 20
Your program:
The guess is too small
Guess 4 of 22:
User: 25
Your program:
The guess is too big
Guess 5 of 22:
User: 23
Your program:
You WIN in 5 guesses – the secret number is 23!

If the player does not guess the number, the following message would appear:
GAME OVER - the secret number is 23!

a) Develop a method called guess. This method has two input parameters: the number of guesses allowed to the user and the secret number to be guessed. The method should return the number of guesses used to find the secret number or the value -1 if the player could not guess the number. It shall prompt the user for guesses and indicate too high or too low with the characters '>' or '<' as shown above. If the user gives a number outside the valid range between 1 and 100, then the error message is displayed and the user is prompted for a valid guess, as shown above.

b) Develop the main method. It should get from the user the level of difficulty before starting the game. In case of invalid values, entered by the user for the difficulty level, it should print an error message as shown above. Any value other than 1, 2, and 3 is invalid for difficulty level (though, you may assume that the user will enter an integer). Using the level of difficulty, determine the number of guesses allowed for the user. Then produce the secret

number to be guessed and call the guess method to let the user play the game. At the end, use the result of the guess method to print the final message.

Note that the Java library has a method Math.random() that generates a random double value in the range [0.0,1.0) (that is greater than or equal to 0.0 and less than 1.0). To generate a random integer value in the range [a,b] (that is, greater than or equal to the value of a or less than or equal to the value of b), you should type-cast the double value to an int value as follows:

```
randomNum = a + (int)(Math.random() * ((b - a) + 1));
```