UNIVERSITY OF ALBERTA

AUCSC 112 LAB

Assignment #2

(Due before midnight on day before the next lab)

Goals:

- Begin to program in the object-oriented style, creating and using an abstract data type (ADT).
- Understand two-dimensional arrays, and nested loop structures.
- Be able to obtain keyboard input from a user.
- Use random numbers.
- Become proficient at Java programming.

Reference:

Heise, Chapter 2.

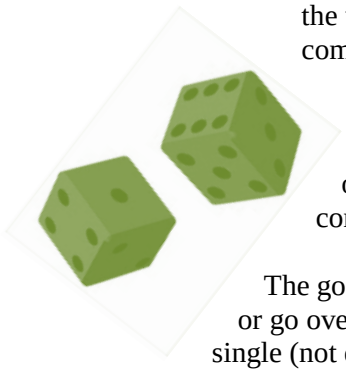
Instructions:

For this week's program you may work as an individual, or a group of two. Do not share any lines of code between groups. If working in a group, each person must write the code together with the other person – do not split the work up and assign parts to each person. One person types, the other thinks, and then switch roles. Your partner must come from your lab section.

Document ALL of your code! Make sure that you add file headers (with your name(s), id #, date, and a brief summary). Also document each of the main steps in your solutions. Ensure that all your variables are named descriptively. All methods should have a header (i.e. documentation explaining the parameters and what each does – use Javadoc format). Avoid documentation that is simply a translation of Java code into English.

The Program to Create:

Construct a Java program that implements a simple dice game. This game involves two players: the user and the computer. The user is always the first player, and the computer plays second.



On a turn, the player rolls the dice once, and continues rolling the dice for as long as they choose or until they roll a single one. The total value on the dice is added to the score for the current round. Once the round is complete then that current round score is added to the total score.

The goal is to be the first player to get to a total of 60 points, or go over 60 points. When a player rolls the dice and gets a single (not double) one, their turn is over and they score 0 for that round. Otherwise, a player may keep rolling and accumulating more for their current score. When a player chooses to stop rolling, the score that has accumulated in the current round is added to their overall score. If a player rolls doubles, then double the value is added to the current score and the player *must* roll again (there is no choice).

Your program must alternate between the user and the computer for turns. The user should be prompted to determine whether they will roll again. The computer should make that choice at random, with a 2/3 chance of saying "roll again" and a 1/3 chance of saying "stay."

You must construct two classes (and therefore have two files within one project): one that represents the two dice (called `TwoDice`) and one for the main game code (called `Main`).

The TwoDice Class:

This class forms an ADT and should contain the following (and may contain more methods):

1) A representation for the two dice, that is, appropriate instance variables. You must include the value of each dice. Also include a 2D array of characters for each die or one 2D array for the set of dice. This array contains the "visual" of a die or the dice. – storage of characters in memory. For example, a die with the value six may "look" like:

o		o
o		o
o		o

Or

	-	-	-	-	-	-	-	
|		o				o		|
|		o				o		|
|		o				o		|
	-	-	-	-	-	-	-	

2) A constructor. Perhaps more than one constructor.
3) A method (either `makeDieAsArray` or `makeDiceAsArray`) to create the physical storage of a die or the dice, as shown in 1).
4) A method (`toString`) to display the dice to standard output.
5) A method (`roll`) to roll the dice.
6) A method (`getValue`) to return the total on the dice.
7) A Boolean method (`isDoubles`) that returns true exactly when the dice contains doubles and false otherwise.
8) A Boolean method (`hasSingleOne`) that returns true exactly when one of the dice has a one, but false when both dice have a one and false when neither dice has a one.

Java's random library will be necessary in getting randomness with rolling the dice. You should be able to complete the `TwoDice` class, independent of the actual game.

The Main Class:

In this file, you implement the actual game, using an object of `TwoDice`. You should write the code to control the turns, keep track of overall scores, and determine an actual winner. Print appropriate information as the game progresses, including the total score lost (if lost), when doubles occur, whose turn it is, etc. Do not use any "global" variables, that is, variables declared in the spot where instance variables are declared in an ADT. This class should only need local variables.

Handle inputs that may not be simply a 'y' or an 'n'. When you read a line, and if the first char is a 'y', 'Y', 'n' or 'N' then continue and assume the appropriate response. Otherwise, ask the user for new input. For example, an entry of "Yo" is treated as "yes", while an entry of "eyes" would ask the user for new input.

Imports Allowed:

In this project, you may only import the following Java libraries:

```
java.util.Scanner
java.util.Random
```

Expected Output:

Here is a sample of the output expected from your program:

```
==================================
          The Dice Game
  How Much Can You Afford to Lose?
==================================
Roll the dice, accumulating the total to add to your score.
Hit 60 before the computer and you win!
If you roll doubles, you get double the value!
           And you must roll again.
If you roll a one - you are done and lose the current round's total,
           unless it's snake eyes!
 -------     -------
|       |   |       |
|   o   |   |   o   |
|       |   |       |
 -------     -------

Here we go...
--------------------------------------

Your turn
Rolling...
 -------     -------
| o   o |   | o   o |
|   o   |   |   o   |
| o   o |   | o   o |
 -------     -------

Doubles!  Roll again!
Rolling...
 -------     -------
| o   o |   |       |
|       |   |   o   |
| o   o |   |       |
 -------     -------

OH NO...You lost it all!
  You lost: 20
Score:  Player 0; Computer 0
--------------------------------------
Computer's turn
Rolling...
 -------     -------
| o     |   | o     |
|       |   |   o   |
|     o |   |     o |
 -------     -------

Rolling...
 -------     -------
| o   o |   | o   o |
|       |   |   o   |
| o   o |   | o   o |
 -------     -------

Staying
Score:  Player 0; Computer 14
--------------------------------------
Your turn
Rolling...
 -------     -------
| o   o |   | o   o |
|   o   |   |       |
| o   o |   | o   o |
 -------     -------
```

Print a game header with simple instructions. Try to print the dice using the `toString` method on dice with values of both 1.

Doubles is an immediate "roll again".

A single one is not good. The players' current round score becomes 0. If they already had a score from a previous round, that amount remains the same.

Show how much is lost (10 * 2 = 20). Do not include the roll with the single 1.

Use random here to make the computer's decision. 2/3 of the time the computer rolls again.

The computer "stayed" so the score (5 + 9) is added to the total score.

```
Roll Again?  (current score is: 9)  Enter 'y' for yes 'n' for no:
res
Sorry did not recognize your response
Roll Again?  (current score is: 9)  Enter 'y' for yes 'n' for no:
yana
Rolling...
.
.
.
Your turn
Rolling...
 -------     -------
| o     |   | o   o |
|   o   |   |       |
|     o |   | o   o |
 -------     -------

Roll Again?  (current score is: 7)  Enter 'y' for yes 'n' for no:
N
Staying
Score:  Player 42; Computer 20
--------------------------------------
Computer's turn
Rolling...
 -------     -------
| o   o |   | o   o |
|       |   | o   o |
| o   o |   | o   o |
 -------     -------

Staying
Score:  Player 42; Computer 30
--------------------------------------
Your turn
Rolling...
 -------     -------
| o     |   | o     |
|   o   |   |       |
|     o |   |     o |
 -------     -------

Roll Again?  (current score is: 5)  Enter 'y' for yes 'n' for no:
y
Rolling...
 -------     -------
| o   o |   | o   o |
|   o   |   |   o   |
| o   o |   | o   o |
 -------     -------

Doubles!  Roll again!
Rolling...
 -------     -------
| o   o |   | o     |
| o   o |   |       |
| o   o |   |     o |
 -------     -------

Roll Again?  (current score is: 33)  Enter 'y' for yes 'n' for no:
n
Staying
Score:  Player 75; Computer 30

  TaTaTah Drum rollllllll
  The winner is:  You
```

Handle input based on the first letter.

When there is a choice to roll again, give the total that might be lost.

Accept either lower or upper case.

Show the score after each player's turn.

Print a message at the end of the game, showing the winner.

How to Hand In Your Work:

If you worked with a partner, please only have one person submit the program and run files (1 – 5), and then both people submit the file in 6. If you worked as an individual, only submit the first 5 items.

1) `Main.java`
2) `Main.pdf` made from your `Main.java` file
3) `TwoDice.java`
4) `TwoDice.pdf` made from your `TwoDice.java` file
5) A pdf of one complete run of your program. Make sure that this run shows all aspects of the assignment.
6) Only if you worked with a partner: a pdf of the group evaluation form (found on eClass or GitHub).

Practice Questions – Assignment 2 Concepts:

These are self-check questions; do not hand in solutions. These questions are provided so that you can check whether you have learned the concepts we are expecting you to learn. No solutions are provided, and we encourage you to discuss these questions with other students.

1) What is a Java class? State two purposes for a Java class?
2) What is an instance variable?
3) When should instance variables be made private in Java?
4) What is final in Java? Where could you use final in your program?
5) What is the difference between a public and a private method in Java?
6) What does a constructor do?
7) When should there be more than one constructor? What is often "the rule" for coding multiple constructors?
8) How do you make a two-dimensional array?
9) What is the index of the top left location of a two-dimensional array? The bottom right location?
10) How do you get the size of an array?
11) How do you get the size of a column of a two-dimensional array?
12) How is a two-dimensional array traversed?
13) What kind of loop is required to initialize a two-dimensional array so that it is all empty?
14) Why are the random integers you used for rolling the dice or deciding the computer's strategy not truly random?
15) What is Javadoc vs. other documentation?
16) What library is used for getting input from a user?
17) Why do you have to specify the type of the item you want to get from the user?
18) What is an object? How do you make an object?
19) How do you make an object do something?
20) How do you obtain the value of a data field in an object?
21) What is the Java code to set the middle row of a two-dimensional character array to all asterisks?
22) What happens if your array index is out of bounds? Give an example of an out-of-bounds array index in your dice game.
23) Do you have more than one method in your `Main` file? Why or why not?
24) When does a method need a static modifier and when should a method not have a static modifier?
25) What does a variable that is all capitals tend to signify? Where did (or should) you use such a variable in your program?

26) Did you use a do-while loop anywhere? If so, what behavior did you want?
27) How is indexing into a String different from indexing into an array?
28) Why should documentation not be an afterthought?
29) Suppose we want to play the game to 100. Where would you change your code? (How many instructions did you change?) Was the number 60 "magic" or did you assign it to a constant?

Image Credits:
Dice: https://clipart-library.com/pictures-of-dice.html
60: https://www.freepik.com/free-photos-vectors/number-60/2#uuid=017a1b6d-39e5-43f8-9ae8-99f554f6f183
Question Mark: https://www.freepik.com/free-photos-vectors/question-mark-cartoon
Hands in: https://www.shutterstock.com/search/cartoon-high-five-hand