

CST8110 - Introduction to Programming

Assignment #3 - Loops

DUE: Friday November 20th at 6pm SHARP - see submission instructions below. Late submissions receive a grade of 0.

Problem Description:

Using the steps for Problem Solving - generate for the following problem:

- a) test plan AND
 - b) write and test the program code in Java.
- This problem will simulate a game of Solitaire Dice (an invented game). You are welcome to customize the game to your own rules - just be sure to document them. **However, you must meet the basic requirements which follow.**
 - The user will start with a pot of money of \$50.
 - In each play of the game, the player will
 - Enter a valid bet amount (less than or equal to the current pot, not negative, a bet of **0 means quit the game**).
 - The game will then remove their bet amount from the pot
 - The game will then roll three die (simulate this using the Random class in Java) display the values and add these values to a total.
 - Then, the game should adjust the pot in the following way:
 - If the total of the three dice is greater than 12, then the player wins their bet back.
 - If two of the three die have the same value, then the player wins double their bet back.
 - If all three die have the same value, then the player wins triple their bet back.
 - Otherwise, the player has lost their money.
 - **The game will end with a bet of 0 OR when the pot reaches 0.**
 - Design of this solution will involve three classes:
 - **Die** class - this class will represent ONE die - with a field of `dieValue`, a constructor, methods `rollDie()`(which gets random value as rolled value and returns it) and `displayDie()`
 - **Game** class - this class will represent the game- with fields of `potAmount` and `betAmount` and a constructor, methods `getBetAmountFromUser()`, and `playGame()`.Note the `playGame()` method will have three local reference variables referring to three different objects of Die class to represent the three dice in the game.
 - **Assign3** class - this class will be the "driver" class and have method `main` which will create an object of Game class, and execute the `playGame()` method

Sample Output : (blue indicates user entered information)

Welcome to Solitaire Dice Game. **Bet an amount** - if the sum of the three die is greater than 12, you keep your bet, if you roll doubles you win double your bet, if you roll triples you win triple your bet, otherwise you lose your bet. A bet of 0 ends the game.

```
Your current pot is 50
Enter your bet amount: 10
Your die are: 3 and 6 and 5
You WIN...your bet back
```

```
Your current pot is 50
Assignment #3 - Loops - Dice Game
```

Enter your bet amount: 10
Your die are: 3 and 6 and 6
You WIN....double your bet

Your current pot is 60
Enter your bet amount: 10
Your die are: 6 and 4 and 6
You WIN....double your bet

Your current pot is 70
Enter your bet amount: 100
Error - cannot bet less than 0 or more than 70...Enter your bet amount: 60
Your die are: 1 and 6 and 2
You LOSE....your bet

Your current pot is 10
Enter your bet amount: 0
You end the game with pot of 10

Submission Requirements:

- You must create a .zip file that contains ONLY the following:
 - Your program code - .java files - (with your name, section, lab teacher listed in comments in the header of each class)
 - A document created with either Notepad, Wordpad or Word named Assign3.docx or Assign3.txt with your test plan - note this should contain your name, section and lab teacher listed at the top.
- The .zip file must have the following as it's name
 - Your last name, your first name, the word assign and the assign number ...
Example CraneLindaAssign3.zip
- Submit the .zip file through the Assignment feature which has been enabled in the CST8110 Blackboard course. This should be directly under the Assignment description.
- Marks will be given for correct submission (ie marks will be deducted for incorrect submission!

Notes on Using the Random class in Java

- See also pages 209-215 of Java, How to Program Textbook
- The Random class is found in library **java.util**.
 - ie ... we need to **import java.util.Random** into a java program to be able to use it
- You need to declare an object of the class first:
 - ie ... declare once in your program, at the beginning:
Random randomNumbers = new Random ();
- Then, you can call a method called **nextInt** on your object of Random class with an integer parameter which will return a random number between 0 and the integer parameter - 1. In other words, the nextInt method returns a random number % the parameter you sent to it.
 - In our case, we want to generate a random number between 1 and 6, so we can execute nextInt(6)...which returns a random number between 0 and 5 and then add 1 to get a number between 1 and 6.
 - ie ... any time you want to generate a random number and store it into a variable that you have already declared (in this example, dieValue), execute :
dieValue = randomNumbers.nextInt (6) + 1;