AP Computer Science Programming Project - Craps

ASSIGNMENT OVERVIEW

In this assignment you'll be creating a main program called `Craps.java`, which will allow the user to play the game of craps on the computer. The program will make use of at least one helper class, `Die.java`.

This assignment is worth 50 points and is due on the *crashwhite.polytechnic.org* server at 23:59:59 on the date given in class.

BACKGROUND

The basic rules for playing craps are relatively simple:

1. A player rolls two six-sided dice and adds the numbers rolled together.
2. On this first roll, a 7 or an 11 automatically wins, and a 2, 3, or 12 automatically loses, and play is over. If a 4, 5, 6, 8, 9, or 10 are rolled on this first roll, that number becomes the "point."
3. The player continues to roll the two dice again until one of two things happens: either they roll the "point" from that first roll again, in which case they win; or they roll a 7, in which case they lose.

Playing craps can include a number of variations on this game, and also typically involves betting on various outcomes; those aspects of the game are not covered in this assignment.

PROGRAM SPECIFICATION

Create a Java program, with associated classes, that:

a. asks the user if they'd like to play craps
b. asks the user if they need instructions, and provide them if desired
c. requires the user to press the <Enter> key each time they want to roll the dice
d. provides status information on the user's *roll*, the *point*, and whether the user has won or lost
e. asks the user if they'd like to play again
f. includes JavaDoc comments, supplemented with additional comments before submitting.

DELIVERABLES

craps.zip

This single file will be a zipped directory (folder) of your BlueJ project. It will include as a minimum your `Craps.java` main program, your `Die.java` class, any other classes you create during the development of your program), and a `package.BlueJ` file.

To submit your assignment for grading, copy your file to your directory in `/home/studentID/forInstructor/` at *crashwhite.polytechnic.org* before the deadline.

ASSIGNMENT NOTES

- This program will require the use of a number of `while` loops, which will repeat rolling the dice until the game is over, and repeat playing the game until the user doesn't want to play anymore.

- To simulate rolling the dice, the program will use the `Math.random()` method.

- Although the program could just repeatedly roll dice for the user without any prompting, part of the fun of playing craps is rolling the dice. The closest we can come to that in this program is asking the user to hit the <Enter> key to roll the dice. This can be accomplished by inserting the following statement into your program at the appropriate location:

  ```
  System.out.print("Press <Enter> to roll...");
  String pause = in.nextLine();
  ```

  We don't actually care about the value of anything the user enters, so the contents of the variable pause aren't important to us.

- One of the main challenges of getting user input is making it appropriately easy: you want your program to have usability. So when the program asks the user if they'd like instructions, what kind of answer might they give? "Yes"? "yes"? Or just a simple "y"? The statements below go a long way towards solving this problem:

  ```
  System.out.print("Would you like to play Craps (Y/n)?");
  String playGame = in.nextLine();
  if (playGame.substring(0,1).equalsIgnoreCase("y"))
  {
        .
        .
  }
  ```

  These statements include aspects of Java that we've already discussed. `in.nextLine()` uses the Scanner `in` that we've established to get a line of input. `playGame.substring(0,1)` refers to the first character in the String `playGame`, so whether the user enters "Yes" or "Y", the program will just look at the first character. The method `.equalsIgnoreCase()` allows us to automatically check for both upper and lowercase situations. Combined, these statements effectively say, "Get the user's input, and if he or she has typed anything that has a first letter 'y' in it, upper or lower case, assume they mean "yes" and perform the statements in the curly braces.

- One additional convention will make this program even more usable. You're probably familiar with the concept of a *default*, something that is assumed to be the case unless otherwise specified. In this program, although we want to ask the user if they want to play, it's highly likely that they DO want to play–they are running the program, after all. So let's make playing the default value: unless they specifically say "No", we're going to assume they want to play. This is subtly indicated in the user prompt–the (Y/n) part–by capitalizing the value that will be assumed if the user just hits the <Enter> key.

  For our program to be able to recognize this entered value of `""` (a pair of quotes with nothing between them), we have to add one more line to the program:

  ```
  System.out.print("Would you instructions on how to play Craps (Y/n)?");
  String playGame = in.nextLine();
  if (playGame.equals("") ||
      playGame.substring(0,1).equalsIgnoreCase("y"))
  {
        .
        .
  }
  ```

GETTING STARTED

1. With paper and pencil, and perhaps in collaboration with a partner, identify what the main components are that you'll need to include in your program.

2. Sketch out the basic flow of your program using a flowchart, and write some pseudocode that you can use to begin implementing those main components.

3. Create a new project in BlueJ that will allow you to manage this assignment.

4. Create a `Die.java` class that you will use to construct and manipulate the dice rolling.

5. Enter pseudocode as comments in the editor, and then fill in more details for various parts of the code.

6. Test each bit of code as you go, making sure that one piece works before you proceed on to the next section. You'll repeatedly run through this edit-compile-test, edit-compile-test process to progressively find bugs and fix them *while* you're writing your program, not afterwards.

7. Save your program from time to time.

8. Once a day or so, archive/zip your BlueJ Craps folder and save a backup copy of it on another device or machine: a flash drive, your home folder on the *crashwhite.polytechnic.org* server, etc.

9. When your program is completed (but before the deadline), copy a final archived package to the server as indicated above.

QUESTIONS FOR YOU TO CONSIDER (NOT HAND IN)

1. Did you use nested `if-else` statements or `if-else if-else` in determining whether the user wins, loses, or has a point to make? Which technique seems more intuitive to you?

2. Is your program "bulletproof," i.e. will it continue to function without any runtime errors, regardless of what the user enters? We'll be looking at more sophisticated ways to prevent different types of errors in your programs in the future.

SAMPLE INTERACTIONS

```
LET'S PLAY CRAPS!
Do you need instructions (Y/n)?
1. Roll two six-sided dice.
2. a. On first roll, if you get a 7 or 11 you win!
2. b. On first roll, if you get a 2, 3, or 12 you lose!
2. c. Any other number you don't win or lose. The die roll becomes your 'point.'
3. Keep rolling the dice again until:
4. a. You roll the point again and win!
4. b. or you roll a 7 and lose.

Good luck!

Press <Enter> to roll the dice...
Your first roll is: 6
That's your point.
Let's see if you can roll it again before you roll a 7!
Press <Enter> to roll the dice...
You rolled: 3
Keep rolling...
Press <Enter> to roll the dice...
You rolled: 6
You rolled your point! You won!
Play again (Y/n)?
Press <Enter> to roll the dice...
Your first roll is: 5
That's your point.
Let's see if you can roll it again before you roll a 7!
Press <Enter> to roll the dice...
You rolled: 8
Keep rolling...
Press <Enter> to roll the dice...
You rolled: 7
You rolled a 7! You lose!
```