Preparing for Lab 5

Prof. Christopher R. Merlo

Dept. of Mathematics, Computer Science, and Information Technology
Nassau Community College
Garden City, NY

CSC 120 CA, Summer I 2016

Getting Started

- Start Eclipse
- Create a new project called lab5
- Create two new classes called Dice and Lab5App
  - Be sure to specify the package lab5
- Write JavaDoc comments at the top of each class
  - Title
  - Description
  - @author Tag

Instance Variables

- Instance variables go in the data-type class
    - In this assignment, that's the `Dice` class
- An object of the `Dice` class represents *three dice*
    - Declare one instance variable for each of the three dice
    - Each variable will store the face value (between 1 and 6 inclusive) of one die
    - Make sure to mark these variables `private`

First Methods

- Write a **default constructor** for the `Dice` class that sets each face value to 0
  - Remember what the name of a constructor must be
  - Remember how many parameters a *default constructor* should have
- Write a `toString` method that returns a reference to a `String` containing the state of the object
  - The format should be: `0 0 0`

JavaDoc

- Write JavaDoc comments for the default constructor and the toString methods
- The format is:

```
/**
 * method name
 * method description
 * @param parameterName1 description of this parameter
 * @param parameterName2 description of this parameter
 * (more @param tags as necessary)
 * @return a description of the return value, if any
 */
```

Testing Your Code So Far

- Testing code goes in the application class `Lab5App`
- Create a `Dice` object
- Display its state to the screen, with the label “After instantiation: ”
- You should get `After instantiation: 0 0 0`

Accessor Methods

- Write an **accessor method** for the first instance variable.
- Be sure to name this method using proper Java naming conventions
  - Accessor method names should begin with “get”
  - The rest of the name should be the variable’s name, capitalized
- Write a JavaDoc comment for this method
- Test the method by calling it from the application class
- Display the result with the label “Value of die 1: ”

`roll` Method

- Write an instance method called `roll()` that rolls all of the dice
- You will need to instantiate the `Random` class and call this instance's `nextInt( bound )` method
- Given `Random random = new Random();`, calling `random.nextInt( 6 )` returns a pseudorandom integer between 0 and 5 inclusive
  - See https://docs.oracle.com/javase/8/docs/api/java/util/Random.html
- Write a JavaDoc comment for the `roll` method
- Call the `roll` method, and then test it by displaying the new state of the `Dice` object
  - `After rolling the dice:  2 4 1`