CMPU 102 Computer Science II  
Lab 02

1. Set up your project folder: Create a folder on the desktop to hold your work for this lab. Rename the folder to cs102-lab02-ellman (but use your last name, rather than mine). Open the CMPU 102 Class Wiki in a browser. Find the Files link for this lab. Download the cmpu102-lab02-files.zip file to your desktop folder; extract all the files; and put them in the new folder you created for this lab.

2. The folder contains a project called “DrawRandomFace”. Open this project in NetBeans and run it. The program draws a vaguely human face. Run the program a few more times. Notice that it always draws the same face. In this lab, you will modify the program so that it makes random variations in the face. You will make extensive use of helper methods in your implementation. You will also learn some new flow-of-control constructs.

3. Create a Utility class: Invoke the menu item: File-NewFile; select Categories: Java and FileTypes: Java Class and LMB click Next. Enter Utility as the class name; leave the package set to drawrandomface; and click Finish. This class will provide tools or generating random information to control the appearance of the face drawing.

4. Implement a method to simulate a coin toss in the Utility class. First import a Java library for generating random numbers by inserting the following statement after the package declaration at the top of the file: “import java.util.Random;” (without the quotes). Now define a public static boolean coinToss() method. The method will operate by generating a random int, and then returning true or false depending on whether the int is even or odd. First define a local variable called “randomInts” of type Random in the body of the method and give it the initial value: new Random();. This variable holds an object that functions as a box of random integers. Each time you invoke randomInts.nextInt(), you get a new random integer. Now finish coinToss method: (1) Generate and store a random integer in a local variable called “face”. (2) Use int Math.abs(int integer) to compute the absolute value of face. (3) Use the modulus operator % to compute and store the remainder when face is divided by two. (4) Finally return the expression remainder==1, which is true when remainder is one (odd) and false when the remainder is zero (even).

5. Modify the Ear and Mouth class constructors to introduce randomness into these shapes by using the coinToss method of the Utility class.

   a. In the Ear class constructor, at the end of the constructor method, declare a new local variable earRing of class GOval. Next add a conditional statement: if (Utility.coinToss()) { ... }. In the body of the conditional, insert statements to create a GOval shape in a suitable location; assign it to the earRing variable; set its filled property; set its color; and add it to the ear drawing.
b. In the **Mouth** class constructor, re-implement the method body to use the **coinToss** method along with a conditional statement to randomly generate either a smile or a frown. First should define a local variable **mouth** of type **GArc** and give it an initial value of **null**. Next add a conditional statement: if (Utility.coinToss()) {
    ...
} else {
    ...
}, and place statements in the empty blocks to generate and store **GArcs** representing either a smile or a frown. Finish the method by setting the color of the **GArc** and adding it to the **mouth**.

6. Implement a method to simulate a die roll in the **Utility** class. Define a **public static int dieRoll(int numSides)** method. The method will operate by generating a random **int**, and then returning the remainder when the **int** is divided by **numSides**. First generate a random **int**, as you did in the **coinToss** method above and store it in a local variable **side**. Next use **int Math.abs(int integer)** to compute the absolute value of **side**. Next use the modulus operator % to compute the remainder when **side** is divided by **numSides**. Finally, return the value of the remainder, which will be a value in the range 0...numSides-1.

7. Implement a method in the **Utility** class to generate a random **Color**. Implement a **public static Color randomColor()** method that returns a random selection from among eight different colors. The method should define a local variable **color** of type **Color** and give it an initial value of **null**. It should then call **dieRoll** in a switch statement of the following form:

    switch (dieRoll(8)) {
        case 0: color = Color.BLACK; break;
        case 1: color = Color.BLUE; break;
        case 2: color = Color.CYAN; break;
        // ... Omitted ...
        case 7: color = Color.WHITE; break;
        default: color = Color.BLACK;
    }

After the **switch** statement, the **randomColor** method should return the **Color** stored in the color variable.

8. Modify the **Face** class to use **randomColor** to generate the colors of the component objects. (But leave the color of the head unchanged. This is important so that the other objects can be seen when placed on top of the head object.) In order to randomize the colors, you need only modify definitions of the color constants appearing at the top of the **Face** class definitions.

9. Implement a method in the **Utility** class to generate a random **double**. Define a **public static double randomDouble(double lBound, double uBound)** method to return a random **double** value between **lBound** and **uBound**. First use the **Math.random()** expression to generate a random number between zero and one. Then use simple arithmetic to transform this value into a number between **lBound** and **uBound**.
10. Modify the **DrawRandomFace** project to use **randomDouble** to generate some of the size and shape parameters of the component objects. For each size/shape parameter you want to randomize, you should choose the **lBound** and **uBound** parameters of the **randomDouble** method to define a range surrounding the fixed values presently in the program.

11. If you have time, modify the program to put spectacles on the face (or not, depending on a coin toss).