Lab 9  Name:________________________  Checked:_____

Objectives:
Practice using **switch** statements; **do** and **for** loops; and explore the use of dialog boxes (JOptionPane)

A. Practice using for-loops
Write a program to input 10 positive numbers and print their maximum. Use a for-loop.

B. Practice using dialog boxes and do loops
Run **EvenOdd.java** and get familiar with how it works. Modify it so that it performs the function of the GPA calculator (from Lab 4b – version that asks each time whether to repeat). You will need to modify the prompts and other interaction (for example, "Enter quality points: " instead of "Enter an integer: ") and to input an additional number for the credits.

C. Practice using switch statements
Write a program to input a number \( n \) that symbolizes a version of the Mac OS X 10.\( n \) software. Output the name of that version of the Mac OS X 10.\( n \) software. For example, if the user inputs 8, then the program should output "Mountain Lion."
[You will need to do a little googling to find the names of the different mac software. They run from 1-10 and most of them are cats, but more recently they are joined by mountains.]

D. Implement a simple experimental setup that uses timings
Download and run **ResponseTimeExperiment.java** (shown on next page)
- Run the program a few times to observe its function.
- Note the use of **System.currentTimeMillis()** to find the current time at two different points during the run, and thus obtain the amount of time elapsed between the issuing of a question and the input of the answer from the user.
- Refer to the code shown on next page. Modify the code enclosed in the box so that it uses the conditional operator to compute the value of **outcome**. Test your program to ensure that it functions in exactly the same way.
- Modify the program so as to:
  - Repeat the experiment 4 times, using a for-loop.
  - Compute the number of correct answers and the average response time of the user; prints these results at the end.
// ResponseTimeExperiment.java Measure response time for addition problems.
// M A Papalaskari

import java.util.Scanner;
import java.util.Random;

public class ResponseTimeExperiment
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        Random rand = new Random();

        System.out.print("Please enter your name: ");
        String name = in.nextLine();

        System.out.println("Hello "+ name
             + " . Please answer as fast as you can."
             + "\n\nHit <ENTER> when ready for the question.");
in.nextLine();

        int a = rand.nextInt(100);
        int b = rand.nextInt(100);

        long startTime = System.currentTimeMillis();
        System.out.print(a + " + " + b + " = ");
        String response = in.nextLine();
        int number = Integer.parseInt(response);
        long endTime = System.currentTimeMillis();

        String outcome;
        if  (number == a + b)
            outcome = "Correct!";
        else
            outcome = "Incorrect."
;
        long reactionTime = endTime - startTime;

        System.out.println(outcome);
        System.out.println("That took " + reactionTime + " milliseconds");
        System.out.println("Thank you " + name + ", goodbye.");
    }
}
Lab 9 Comments

Comments on this lab, please:

What was the most valuable thing you learned in this lab?

What did you like best about this lab?

Was there any particular problem?

Do you have any suggestions for improving this lab as an effective learning experience?