

Lab 4 – Loops (16 pts)

Lab Objectives

- Be able to convert an algorithm using control structures into Java code
- Be able to write a `while` loop
- Be able to write a `do-while` loop
- Be able to write a `for` loop
- Be able to use the `Random` class to generate random numbers

Deliverables

~~This lab has three tasks. When you have all tasks done, run the report in Blackboard. The report is a Blackboard test with short-answer, file-response, multiple-answer, and other types of questions.~~

~~In the report, you will provide a screenshot of program execution, Java source code files (must have extension `.java`), and your analysis of the results.~~

~~Screenshots in your report must show a full screen, so your computer can be identified. Please resize your IDE panels the way that the required dialog or output is visible along with the source code. Show as much source code as possible.~~

NOTE:

- ~~Use **Blackboard only** to submit your work; **no email** submission unless your instructor directs it.~~
- ~~If Blackboard gives you multiple submission attempts (usually three), the **last one** will be evaluated and graded.~~
- ~~**No late submissions, no changes** in your submission after the due date.~~

Introduction

This is a simulation of rolling dice. Actual results approach theory only when the sample size is large. So, we will need to repeat rolling the dice a large number of times (we will use 10,000). The theoretical probability of rolling doubles of a specific number is 1 out of 36 or approximately 278 out of 10,000 times that you roll the pair of dice. Since this is a simulation, the numbers will vary a little each time you run it.

Check out how to use the random number generator to get a number between 1 and 6 to create the simulation. The concept is introduced in Section 4.11 of the text and commented at the end of the second lecture on loops (see slides).

We will continue to use control structures that we have already learned while exploring control structures used for repetition. We shall also continue our work with algorithms by translating a given algorithm into Java code in order to complete our program. We will start with a `while` loop, then use the same program, changing the `while` loop to a `do-while` loop and then a `for` loop.

Task #1 The `while` Loop (8 pts)

1. Copy the file *DiceSimulation.java* as directed by your instructor. Correct syntax errors, if any, and improve programming style when necessary (indents, newlines, etc.). *DiceSimulation.java* is incomplete. Since there is a large part of the program missing, the output will be incorrect if you run *DiceSimulation.java*.
2. We have declared all the variables. You need to add code to simulate rolling the dice and keeping track of the doubles. Convert the algorithm below into Java code and place it in the `main` method after the variable declarations, but before the output statements. You will be using several control structures: a `while` loop and an `if-else-if` statement nested inside another `if` statement. Use the indenting of the algorithm to help you decide what is included in the loop, what is included in the `if` statement, and what is included in the nested `if-else-if` statement.
3. To “roll” the dice, use the `nextInt` method of the random number generator to generate an integer from 1 to 6.

Repeat while the number of dice rolls are less than the number of times the dice should be rolled.

Get the value of the first die by “rolling” the first die

Get the value of the second die by “rolling” the second die

If the value of the first die is the same as the value of the second die

If value of first die is 1

Increment the number of times snake eyes were rolled

Else if value of the first die is 2

Increment the number of times twos were rolled

Else if value of the first die is 3

Increment the number of times threes were rolled

Else if value of the first die is 4

Increment the number of times fours were rolled

Else if value of the first die is 5

Increment the number of times fives were rolled

Else if value of the first die is 6

Increment the number of times sixes were rolled

Increment the number of times the dice were rolled

4. Compile and run. You should get numbers that are somewhat close to 278 for each of the different pairs of doubles. Run it several times. You should get different results than the first time, but again it should be somewhat close to 278.

Task #2 Using Other Types of Loops: do-while (4 pts)

1. Change the `while` loop to a **`do-while`** loop.
2. Make other necessary changes to save your work as new file named `DiceSimulation_Do.java`.
3. Compile and run. You should get the same results as at Task #1.

Task #3 Using Other Types of Loops: for (4 pts)

1. Change the `do-while` loop to a **`for`** loop.
2. Make other necessary changes to save your work as a new file named `DiceSimulation_For.java`.
3. Compile and run. You should get the same results as in Task #1.