

Project 5

CS 1323, Fall 2021

Learning Objectives

1. Use a nested if statement in a program. (30 points)
2. Generate random numbers between 1 and 10 randomly. (10 points)
3. Use logical operators (&&, ||, !) in a program. (10 points)
4. Implement the logic of a simple slot machine correctly. (40 points)

10 points will be awarded for the documentation of your program. That means using good names for variables, proper and consistent indentation of code, sufficient comments and meaningful use of whitespace.

1324: When your program is completed and running, have the teaching assistants check it to get credit for the lab. If you do not complete the laboratory during the allotted time, you may submit it on Zybooks before Monday, October 4 at 11:59 p.m. Only people who attend the whole laboratory and actively participate in the development of code with their assigned partner will be permitted to submit assignments for credit.

1323: Submit your program on Zybooks before Monday, October 4 at 11:59 p.m.

Description

In this laboratory you will write a simplified version of a slot machine. Although physical slot machines can be used for gambling, we are not gambling in this class.

Here is how a simple slot machine works: Three numbers between 1 and 10 are drawn randomly. If all three numbers match you win 50 points. If the numbers are all 7, you win 500 points. If two numbers match, but not three, you win 3 points. If you have two 7s (but not three 7s), then you win 15 points. If no numbers match, then you lose one point.

Four example runs of the program are below.

Run #1:

Your spin was 5 9 1.
You earned -1 points.

Run #2:

Your spin was 4 3 4.
You earned 3 points.

Run #3:

Your spin was 7 3 7.
You earned 15 points.

Run #4:

Your spin was 5 5 5.
You earned 50 points.

Generating Random Numbers

The Math class contains a method that will generate a random double between 0 (inclusive) and 1 (exclusive). This method should be used to create random numbers for the project.

Suppose that we were trying to create random numbers between 6 and 9 (both inclusive). We need to find mathematical operations that will manipulate the Math.random() value between 0 and 1 into this new range.

The table below summarizes the calculation. By looking at what happens to the smallest, middle, and largest values we can make sure the equation is doing the right things. I multiplied the generated random number by 4 because I need one of four values (0, 1, 2, 3) to add to 6.

| | Math.random | Math.random()*4 | (int) (Math.random() * 4) | (int) (Math.random() * 4) + 6 |
|---------|-------------|-----------------|---------------------------|-------------------------------|
| Minimum | 0.0 | 0.0 | 0 | 6 |
| Middle | 0.5 | 2.0 | 2 | 8 |
| Maximum | .999999 | 3.99999 | 3 | 9 |

You'll need to modify the 4 and 6 in the table below to generate numbers between 1 and 10 (inclusive). The editable table below can be used to help you figure out how this should work. Pay attention to precedence.

| | Math.random | | | |
|---------|-------------|--|--|--|
| Minimum | 0.0 | | | |
| Middle | 0.5 | | | |
| Maximum | .999999 | | | |

Ordering of Logic

The logic in the program can be implemented as a cascading if statement. The order in which you identify the cases will be important. The problem occurs because, for example, a triple of 5s also contains a pair of 5s. So if you identify pairs before triples, you will never detect triples. This will be hard to spot in your program because triples (and especially triple 7s) are rare. As a general rule in these circumstances, it is best to move from the most specific case to the least specific case.

Multiple Runs

To make sure that your program is running correctly, you will need to run it many times. You need to make sure that 1 and 10 both occur as random numbers, and that no other values occur. You also need to make sure that the right thing is being done in each case. This is hard because three matching numbers are relatively rare (probability: 0.01) and getting three matching 7s is even more rare (probability 0.001).