Name _____ Period _____

Math Class

Your Tasks (Mark these off as you go)

- ☐ Explore the `random()` method from the Java Math class
- ☐ Explore binary numbers
- ☐ Explore how computers store floating point numbers in binary
- ☐ Receive credit for this lab guide

☐ Explore the random() method from the Java Math class

Previously we learned that we can use the `random()` method from the Java Math class to create a random number from 0 up to 1, where 1 is not inclusive. Below is an example,

Code	Output

`System.out.println(Math.random());`
`System.out.println(Math.random());`
`System.out.println(Math.random());`
`System.out.println(Math.random());`
`System.out.println(Math.random());`
	`0.630453986555391`
	`0.9500392413268637`
	`0.5758294866895269`
	`0.2530745277465285`
	`0.7569585160201271`

But, how do we manipulate this method to simulate different scenarios? For example, a dice role, or a random card in a 52-card deck?

The random() method can be scaled by multiplying it by the range of numbers we want to include. For example, if we wanted to simulate a 6-sided die we could multiply 6. If we wanted to simulate a 52 card deck we could multiply by 52. Below, is example of how we can scale the random method to generate numbers from 0 up to 6.

Code	Output

`System.out.println(Math.random()*6);`
`System.out.println(Math.random()*6);`
`System.out.println(Math.random()*6);`
`System.out.println(Math.random()*6);`
`System.out.println(Math.random()*6);`
	`0.6023513008272738`
	`1.4788889884708731`
	`4.718693470120201`
	`1.607710172697613`
	`2.518787436103896`

But, we still have a problem if we want to model a 6-sided die. First, the numbers on a die are integers – we have created doubles. Second, the numbers on a die start at 1, not 0. To generate random integers starting at 1 requires that we shift the range. This is done by adding 1. To create random integers, we cast the final number generated to an int.

Code	Output
`System.out.println((int)(Math.random()*6+1));`	3 5 5 1 6

Write a “lucky” number generator that will create random numbers between -100 and 100 (100 not inclusive). The output to the console should be displayed in a single line separated by commas exactly as shown (your numbers will obviously be different than my numbers ;-)). My lucky numbers are: 10, 66, 42

Write to code to simulate a coin flip. You code should print the number of heads after 5 flips. No, *if* statements allowed 😉

□ Explore binary numbers

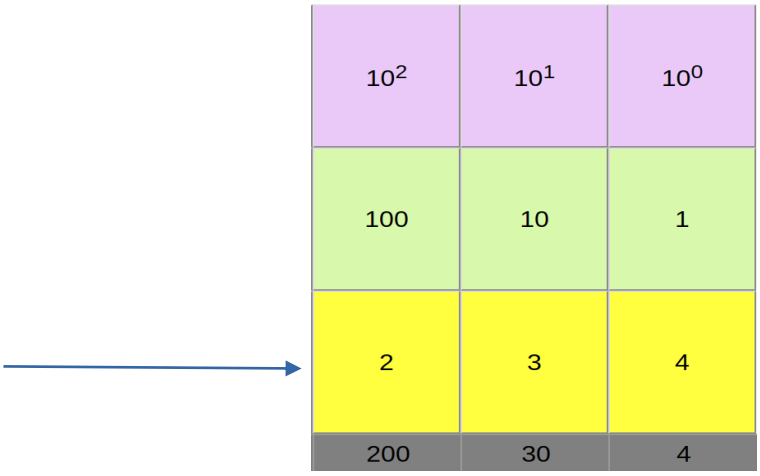
As humans, we typically represent numbers in the decimal system. Counting to ten is as simple as 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

As we just learned, computers represent all information in bits. In order to represent numbers with just 0s and 1s computers use the binary number system. Here's what it looks like when a computer counts to ten: 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010.

Decimal Number Refresher

Before exploring how the binary system works, let's revisit our old friend, the decimal system. When you learned how to count, you might have learned that the right-most digit is the "ones' place", the next is the "tens' place", the next is the "hundreds' place", etc.

Another way to say that is that the digit in the right-most position is multiplied by 1, the digit one place to the left is multiplied by 10, and the digit two places to its left is multiplied by 100. The 234 can be visualized as follows,



10^2	10^1	10^0
100	10	1
2	3	4
200	30	4

When we multiply each digit by its place, we can see that 234 is equal to,

(2 x 100) + (3 x 10) + (4 x 1).

We can also think of those places in terms of the powers of ten. The ones' place represents multiplying by 10^0, the tens' place represents multiplying by 10^1, and the hundreds' place represents multiplying by 10^2. Each place we add, we are multiplying the digit in that place by the next power of 10.

Binary numbers

The binary system works the same way as decimal. The only difference is that instead of multiplying the digit by a power of 10, we multiply it by a power of 2.

Let's look at the decimal number 1, represented in binary as 0001:

2^2	2^1	2^0
4	2	1
0	0	1
0	0	1

That's the same as (0 x 8) + (0 x 4) + (0 x 2) + (1 x 1), or 0 + 0 + 0 + 1.

Now let's consider the decimal number 10 which is represented in binary as 1010:

2^3	2^2	2^1	2^0
8	4	2	1
1	0	1	0
8	0	2	1

That's the same as (1 x 8) + (0 x 4) + (1 x 2) + (0 x 1), or 8 + 0 + 2 + 0.

To get some practice converting binary numbers to decimal, I've created a virtual "flippy do". Click on the link below, the use the flippy do to figure out the decimal equivalents of the following binary numbers: 1110, 110011, 10001 https://flippydo.hpluska.repl.co/

Now that you are familiar with the mechanics of converting a binary number to decimal. Write code that could be used to convert a four-digit binary integer to decimal. You will need to utilize the operations you are already familiar with in conjunction with the pow method from the Java Math class.

☐ Explore how computers store floating point numbers in binary

In addition to integers, computers are also capable of storing floating point numbers. Consider a new version of the flippy-do below. Notice that decimals can also be stored in binary by using negative exponents. Using this version of the flipping-do, 0.1 in binary is 0.5 in decimal. And, .101 in binary is 0.625 in decimal.

2^0	2^{-1}	2^{-2}	2^{-3}
1	0.5	0.25	0.125
0	1	0	1
	.5	0	.125

If 2^{-1} is .5 and 2^{-2} is 0.25, what are the values of 2^{-3}, 2^{-4}, 2^{-5}?

Use the flippy-do 2, https://flippydo2.hpluska.repl.co/ to Convert each of the floating point numbers to binary
0.125
0.03125
0.675
10.75

Now that you are familiar with the mechanics of converting a binary floating point numbers to decimal. Write code that could be used to convert a four-digit binary float number to decimal. You will need to utilize the operations you are already familiar with in conjunction with the pow method from the Java Math class.

□ Receive Credit for this lab guide

Submit this portion of the lab to Pluska to receive credit for the lab guide. Once received, your completed code challenges will also be graded and will count towards your final lab grade.