

Math in Java

Random Numbers

The Importance of Randomness

- modern computers will often attempt to simulate real-life
- the real world has many phenomena that are random, or have an element of chance
- for example,
 - gambling (dice, cards, etc.)
 - weather (movement & interaction of air particles)
 - human responses (as simulated in games)

Generating a Random Number

Math.random()

Java provides a special method called `Math.random()` for producing random numbers.

This method produces a very specific and limited output. It is then up to the programmer to turn that into something more appropriate to their application.

The result is between 0.000000 and 0.999999

```
System.out.println(Math.random());
```

Example: Random Numbers

simulate a six-sided die

A single die has six sides. Assuming it is a fair die, the odds of getting any roll (from 1 to 6) will be the same.

How can we simulate this result using a random number generator from 0.000000 to 0.999999?

Example: Random Numbers

simulate a six-sided die

How can we simulate this result using a random number generator from 0.000000 to 0.999999?

Step 1: Multiply by the number of possible outcomes

A six-sided die has 6 possible outcomes (1, 2, 3, 4, 5, 6).

	<u>Low Value</u>	<u>High Value</u>
Before:	0.000000	0.999999
After:	0.000000	5.999999

Example: Random Numbers

simulate a six-sided die

How can we simulate this result using a random number generator from 0.000000 to 0.999999?

Step 2: Change the result to an integer

So far we have decimal values, but a die is integers. We convert to an integer by chopping off the decimal part (this is NOT rounding... the decimal part is just cut away).

	<u>Low Value</u>	<u>High Value</u>
Before:	0.000000	5.999999
After:	0	5

Example: Random Numbers

simulate a six-sided die

How can we simulate this result using a random number generator from 0.000000 to 0.999999?

Step 3: Add or subtract the starting value

We now have 6 values (0, 1, 2, 3, 4, 5). For our example, we want these values to start at 1, not zero. Add 1 and we should have the correct range of values.

	<u>Low Value</u>	<u>High Value</u>
Before:	0	5
After:	1	6

Example: Random Numbers

simulate a six-sided die

The code to produce this result would be:

```
int dieRoll = (int) ( 6 * Math.random() ) + 1;
```

↑
variable
to store
random
value

↑
convert
to integer
value

↑
number of
possible
outcomes

↑
generate
random
value

↑
add offset
to starting
value

Example: Random Numbers

determine the possible results

Consider the following code:

```
int value = (int) ( 10 * Math.random() ) + 5;
```

↑
variable
to store
random
value

↑
convert
to integer
value

↑
number of
possible
outcomes

↑
generate
random
value

↑
add offset
to starting
value

Example: Random Numbers

determine the possible results

Consider the following code:

```
int value = (int) ( 11 * Math.random() ) - 5;
```

↑
variable
to store
random
value

↑
convert
to integer
value

↑
number of
possible
outcomes

↑
generate
random
value

↑
add offset
to starting
value