Generating Random Double Values

The random() method of the Math class returns a double!

To generate random **double** values between 0 inclusive and 1 exclusive, and store the result in x, what do you do? You just use: double x = Math.random();

When this method is called, a double value is returned in the range: 0.0 <= x < 1.0 so you

can think of that as 0.0 <= x < = 0.999999999999999999…. (1.0 is not included)

So to generate random doubles between 0 inclusive and 10 exclusive, what do you do?

You use: double x = Math.random() \* 10.0; or just double x = Math.random() \* 10;

And to generate random doubles between 0 inclusive and 50 exclusive, what do you do?

You use: double x = Math.random() \* 50.0; or just double x = Math.random() \* 50; 

Generating Random Integers

If you want to generate random **integers**, you must use casting!

Specifically if you want to generate a random integer between 0 inclusive and 10 inclusive:

you use: int y = (int) (Math.random() \* 11) Especially notice the placement of the red ( ).

Note that 11 is one larger than 10. This is needed because casting will truncate (round down). The above line will not work unless you put parentheses around (Math.random() \* 11). We need to be able to cast the entire expression (Math.random() \* 11) to an int using the (int) cast operator. If you wrote the entire line of code like this:

int y = (int) (Math.random() ) \* 11 , then because of order of operations, only Math.random() will be cast to an int. What happens is (int) (Math.random() ) always evaluates to 0!!!

This causes a problem because 0 \* 11 is 0 and no matter what you are multiplying by the

answer will always be 0. So be extra careful when generating random integers and using ( ).

So take note of what range of integers each of the following lines of code generate:

(int) (Math.random() \* 3) would produce 0, 1, or 2

(int) (Math.random() \* 6) would produce 0, 1, 2, 3, 4, or 5

(int) (Math.random() \* 8) would produce 0, 1, 2, 3, 4, 5, 6, or 7

(int) (Math.random() \* 100) would produce 0 .. 99

(int) (Math.random() \* 175) would produce 0 .. 174

If you want random integers between 1 and 10 inclusive, then we need to do the following …

1 10 ← orignal range

- 1 - 1 subtract 1 from both range numbers.

new range → 0 9 multiply by 10 (1 more than 9).

So then you use y = (int) (Math.random() \* 10) + 1 add back the 1 you subtracted.

You might think on first glance that this is generating values in the same range as:

int y = (int) (Math.random() \* 11) ………….. but it isn’t! this range is 0 to 10 not 1 to 10.

More coming up on generating random integers.

More About Generating Random Double Values

To generate random doubles between 30.0 inclusive and 50.0 exclusive, what do you do?

30.0 50.0

- 30.0 - 30.0 subtract 30 from both numbers

new range → 0 20.0 multiply by 20 … NOT 21

then you use Math.random() \* 20.0 + 30.0 add back the 30 you subtracted

(no need for parenthesis because of order of operations or casting because of generating doubles)



To generate random doubles between -20 inclusive and -10 exclusive, what do you do?

-20.0 -10.0

+ 20.0 + 20.0 add 20 to both numbers

new range → 0 10.0 multiply by 10 … NOT 11

then you use Math.random() \* 10.0 – 20.0 subtract off the 20 you added

(no need for parenthesis because of order of operations or casting because of generating doubles)

Keep Going! You’re almost done!

More About Generating Random Integers

If you want to generate random integers between 25 and 100 inclusive, what do you do?

25 100

- 25 - 25 subtract 25 from both numbers

new range → 0 75 multiply by 76 for it to be inclusive

then you use (int) (Math.random() \* 76) + 25 add back the 25 you subtracted

If you want random integers between -75 and -25 inclusive, then …

-75 -25

+ 75 + 75 add 75 to both numbers

new range → 0 50 multiply by 51 for it to be inclusive

then you use (int) (Math.random() \* 51) - 75 subtract off the 75 you added

THE END