

# Lesson 42: Craps Lab #2 (W12D3)

Balboa High School

Michael Ferraro

November 4, 2015

- 1 Create class RandomInts in new project Lesson42.
- 

```
public class RandomInts {  
  
    public static void main(String[] args) {  
        double a = Math.random();  
        System.out.println(a);  
    }  
  
}
```

---

- 2 Enhance the main() so that it prints a random number on the screen exactly 17 times using a for() loop.

# Aim

Students will receive a short lesson on random number generation for use with the *Craps Lab* and have time to work on §5 of PS #6.

# What's `Math.random()`?

- Q: How does `Math.random()` appear to work?

---

<sup>1</sup>Note the right parenthesis after 1.0 — that means the range is *not* inclusive of the value 1.0. So 0.9999999 is a possible return value.

# What's `Math.random()`?

- Q: How does `Math.random()` appear to work?
- A: Returns a double from `[ 0.0 → 1.0 )`<sup>1</sup>

---

<sup>1</sup>Note the right parenthesis after 1.0 — that means the range is *not* inclusive of the value 1.0. So 0.9999999 is a possible return value.

# What's `Math.random()`?

- Q: How does `Math.random()` appear to work?
- A: Returns a double from `[ 0.0 → 1.0 )`<sup>1</sup>
- The `Math.random()` API is [here](#).

---

<sup>1</sup>Note the right parenthesis after 1.0 — that means the range is *not* inclusive of the value 1.0. So 0.9999999 is a possible return value.

# What's `Math.random()`?

- Q: How does `Math.random()` appear to work?
- A: Returns a double from `[ 0.0 → 1.0 )`<sup>1</sup>
- The `Math.random()` API is [here](#).
- Taming the beast: How might we get this method to return an integer between 1 and 8, for example?

## Multiplication and casting!

---

<sup>1</sup>Note the right parenthesis after 1.0 — that means the range is *not* inclusive of the value 1.0. So 0.9999999 is a possible return value.

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...



## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

```
8 * Math.random()
```

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

```
8 * Math.random()
```

```
8 * [ 0.0 → 0.9 ]
```

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

```
8 * Math.random()  
8 * [ 0.0 → 0.9 ]  
[ 0.0 → 7.9 ]
```

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

$$\begin{aligned} &8 * \text{Math.random}() \\ &8 * [ 0.0 \rightarrow 0.\bar{9} ] \\ &[ 0.0 \rightarrow 7.\bar{9} ] \end{aligned}$$

Now cast that result into an `int`:

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

```
8 * Math.random()  
8 * [ 0.0 → 0.9̄ ]  
[ 0.0 → 7.9̄ ]
```

Now cast that result into an `int`:

```
(int)[ 0.0 → 7.9̄ ]
```

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

```
8 * Math.random()  
8 * [ 0.0 → 0.9̄ ]  
[ 0.0 → 7.9̄ ]
```

Now cast that result into an `int`:

```
(int)[ 0.0 → 7.9̄ ]  
[ 0 → 7 ]
```

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

$$\begin{aligned} &8 * \text{Math.random}() \\ &8 * [ 0.0 \rightarrow 0.\bar{9} ] \\ &[ 0.0 \rightarrow 7.\bar{9} ] \end{aligned}$$

Now cast that result into an `int`:

$$\begin{aligned} &(\text{int})[ 0.0 \rightarrow 7.\bar{9} ] \\ &[ 0 \rightarrow 7 ] \end{aligned}$$

So multiplying `Math.random()` by 8 and then casting into an `int` gives 8 discrete values:  $\{0, 1, 2, 3, 4, 5, 6, 7\}$ . If we want the integers  $1 \rightarrow 8$ , add 1:

## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

$$\begin{aligned} &8 * \text{Math.random}() \\ &8 * [ 0.0 \rightarrow 0.\bar{9} ] \\ &[ 0.0 \rightarrow 7.\bar{9} ] \end{aligned}$$

Now cast that result into an `int`:

$$\begin{aligned} &(\text{int})[ 0.0 \rightarrow 7.\bar{9} ] \\ &[ 0 \rightarrow 7 ] \end{aligned}$$

So multiplying `Math.random()` by 8 and then casting into an `int` gives 8 discrete values:  $\{0, 1, 2, 3, 4, 5, 6, 7\}$ . If we want the integers  $1 \rightarrow 8$ , add 1:

$$(\text{int})( 8 * \text{Math.random}() ) + 1$$



## Making `Math.random()` Useful

Consider what would happen to the return value of `Math.random()` if we multiplied by 8...

$$\begin{aligned} &8 * \text{Math.random}() \\ &8 * [ 0.0 \rightarrow 0.\bar{9} ] \\ &[ 0.0 \rightarrow 7.\bar{9} ] \end{aligned}$$

Now cast that result into an `int`:

$$\begin{aligned} &(\text{int})[ 0.0 \rightarrow 7.\bar{9} ] \\ &[ 0 \rightarrow 7 ] \end{aligned}$$

So multiplying `Math.random()` by 8 and then casting into an `int` gives 8 discrete values:  $\{0, 1, 2, 3, 4, 5, 6, 7\}$ . If we want the integers  $1 \rightarrow 8$ , add 1:

$$\begin{aligned} &(\text{int})( 8 * \text{Math.random}() ) + 1 \\ &\text{gives } \{1, 2, 3, 4, 5, 6, 7, 8\} \end{aligned}$$

# You Try!

- This method for getting Java to return random ints will be on the AP exam!
- Modify your Do Now class to produce random ints from 1 to 10, inclusive. If you get stuck, follow the sequence from the last slide. Also, think about how many discrete values you need.

## Yet Another Way...

There's another way to generate random integers: `java.util.Random`

## Yet Another Way...

There's another way to generate random integers: `java.util.Random`

- Add an `import` statement at the top of your class.

## Yet Another Way...

There's another way to generate random integers: `java.util.Random`

- Add an `import` statement at the top of your class.
- Before your `for()` loop, create a `Random` object:  
`Random r = new Random();`

## Yet Another Way...

There's another way to generate random integers: `java.util.Random`

- Add an `import` statement at the top of your class.
- Before your `for()` loop, create a `Random` object:  
`Random r = new Random();`
- Inside your `for()` loop, call `nextInt()`:  
`a = r.nextInt(7);`

## Yet Another Way...

There's another way to generate random integers: `java.util.Random`

- Add an `import` statement at the top of your class.
- Before your `for()` loop, create a `Random` object:  
`Random r = new Random();`
- Inside your `for()` loop, call `nextInt()`:  
`a = r.nextInt(7);`
- See what the range of output values is and adjust that last line of code so that your range is from 1 to 8.

- The rest of the period is yours to work on PS #6
- I'll run the autotester later for the §4 problems so you can find out whether your solutions are OK.



Finish §5 of PS #6 and all prior sections.