

## Java Fundamentals

### 3-5: Randomization and Constructors

### Practice Activities

#### Objectives:

- Create randomized behaviors
- Define comparison operators
- Create if-else control statements
- Create an instance of a class
- Recognize and describe dot notation

#### Vocabulary:

Identify the vocabulary word for each definition below.

	A technique that allows a class to use a method from another class or object. The dot between the class/object name and the method name indicates that the method comes from a different class or object.
	A special kind of method that is automatically executed whenever a new instance of the class is created.
	A keyword that indicates that a new object is being created.
	Symbols that compare two random values in a method.

#### Try It/Solve It:

1. Write the method signature of the method that allows a class to return a random number. Write your response below:
2. State three reasons why random behavior will be an important component of the game you are creating.
3. In the getRandomNumber() method, what does the parameter expect? Write your response below:
4. How can you verify if a method is a static method? Write your response below:
5. Describe what the following method does. Write your response below.  
`Greenfoot.getRandomNumber(15)`
6. Create a table that lists the comparison operators, and what each one does.
7. Express the following statement in source code: We want to program a flower to turn a random number of degrees, up to 15 degrees, 28% of the time.
8. Express the following statement in source code: We want a bumblebee to turn 35 degrees if a random number less than 5 is returned, and if a number greater than 5 is returned, we want the bumblebee to turn 2 degrees.

9. In the following constructor, what size is the world?

```
public BeeWorld()  
{  
    super(550, 300, 1);  
}
```

10. In your scenario's world constructor, write code that automatically creates two new instances of an Actor class when the world is initialized.
11. **Note: Coding from this point will all be done on the same scenario. Use the base scenario - JF\_3\_5\_Practice\_Start as your starting point.**

We are going to create a game called Grab the Barrel.

You are the pilot of a plane that has been sent out to pick up barrels that have fallen off a cargo ship. By flying over a barrel you will automatically collect it. The problem is that sea is rough and the barrels keep going under the water, so you have to collect them when they are at the surface. To make matters worse the area is a rocket testing site for rockets from NASA and they are unable to stop these being fired into the same area that you are in!

You are going to program the above game.

The plane will always be moving but we can control its left and right turning. Flying over a barrel will mean you collect it and search for the next one. Collecting a barrel will give us a score point.

The barrel will randomly appear on the screen, but only stay for a set amount of time and then go under and re-appear randomly elsewhere on the screen.

The rockets will appear at the top of the screen then randomly move down the screen until they disappear off the map.

You must not hit the rocket. It is estimated that you will be able to survive 3 direct rocket hits and then the game will be over.

Open course scenario "JF\_3\_5\_Practice\_Start".

12. Create a sub class of the World class called **BarrelWorld** with the dimensions 800 x 600. Use the scenario image wet-blue.jpg as the scenario image.
13. Create a subclass of Actor called **Plane** and use the image airplane1.
14. Create a subclass of Actor called **Barrel** and use the image Barrel.
15. Create a subclass of Actor called **Rocket** and use the image rocket1.
16. Add code to automatically add a Barrel to the top left.
17. Add code to add a Rocket to the top of the screen. This code will randomly generate the x-coordinate and set the y coordinate to 10.
18. Add code to automatically add the plane instance to the middle of the screen.
19. Add code to the **Plane** class to have it constantly move forward at a speed of 2 and turn left when the left arrow key is pressed or right when the right arrow key is pressed. It will turn 2 degrees.