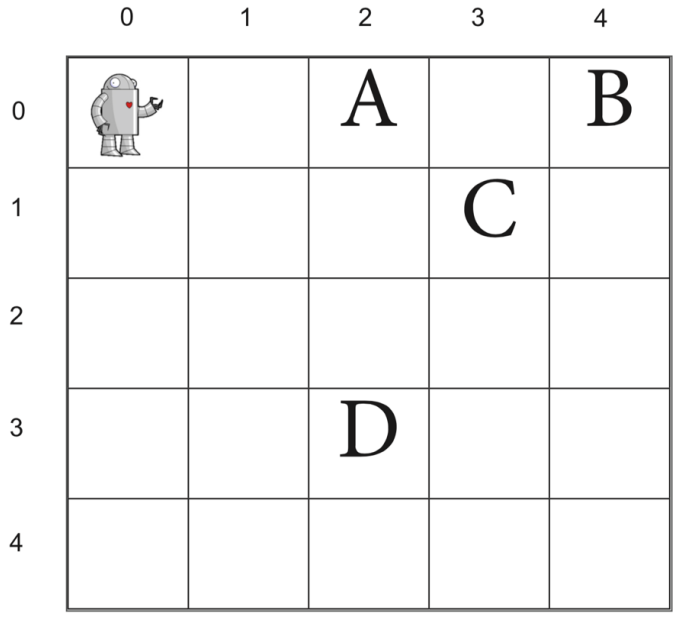
Lecture 2 Clicker Questions:

1. Where will `samBot` end up when this code is executed?

```
samBot.moveForward(3);
samBot.turnRight();
samBot.turnRight();
samBot.moveForward(1);
```



   Answer: A; The first line instructs `samBot` to move forward three spaces, then it tells her to make two right turns. Since she makes those turns in place, the next command to move forward one space puts her in square A.

2. You know that `blueBot` and `pinkBot` are instances of the same class. Let's say that the call `pinkBot.chaChaSlide();` makes `pinkBot` do the cha-cha slide. Which of the following is true?
   A. The call `blueBot.chaChaSlide();` will make `blueBot` do the cha-cha slide
   B. The call `blueBot.chaChaSlide();` might make `blueBot` do the cha-cha slide or another popular line dance instead
   C. You have no guarantee that `blueBot` has the method `chaChaSlide();`

   Answer: A; All instances of a class have the exact same capabilities. For the example of the `Robot` class, each instance has the methods defined in the `Robot` class. Since `pinkBot` and `blueBot` are both instances of `Robot`, if `pinkBot` has the capability to do the cha-cha slide (meaning that it knows the `chaChaSlide();` method) then we know that the call `blueBot.chaChaSlide();` will make `blueBot` do the cha-cha slide.

3. Give this method, what can we say about `this.turnRight()`?

```
public class Robot {
      /* additional code elided */
      public void turnLeft() {
      this.turnRight();
      this.turnRight();
      this.turnRight();
      }
}
```

A. Other objects cannot call the `turnRight()` method on instances of the `Robot` class
B. The current instance of the `Robot` class is calling `turnRight()` on another instance of `Robot`
C. The current instance of the `Robot` class is calling the `turnRight()` method on itself
D. The call `this.turnRight();` will not appear anywhere else in the `Robot`'s class definition

Answer: C; The keyword `this` is how an instance of a class can call a method on itself. In the `turnLeft()` method, this instance of the `Robot` class calls the `turnRight()` method on itself three times in order to perform this command. This does not mean that this cannot be used anywhere else in the code or that other objects cannot call the `turnRight()` method on instances of the `Robot` class. By calling `this.turnRight();`, this instance of the `Robot` class is saying "hey me, turn right!".