Unit 2 - Lesson 3
Parameterized Constructors

Computer Science A

Warm Up

Retrieve
your knowledge and ideas and write it down silently

Pair
up with a neighbor and talk about your reflections

Share
your thoughts in a class discussion

💡 Discuss:

What did you **wish you could do** when creating a `Painter` object?

Activity

🎯 Lesson Objectives

By the end of this lesson, you will be able to . . .

- Write a parameterized constructor to specify formal parameters
- Identify the correct constructor being called based on the constructor signature

🚀 Question of the Day

How can I create objects with specific values for its instance variables?

Predict and Run

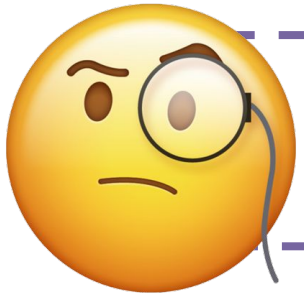
Navigate to Lesson 3, Level 1

✅ Do This:

1. Predict the output of the program

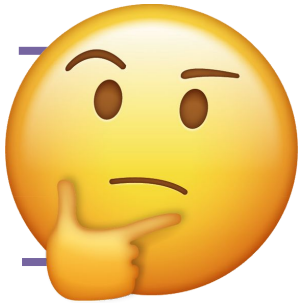
   There are no wrong answers!

2. Run it to compare your prediction with the results

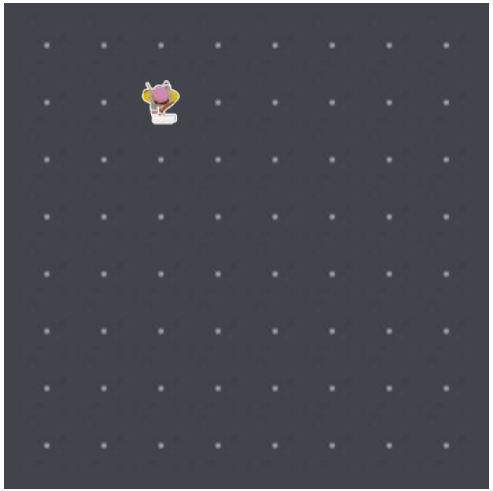


What did you notice about the code in this program?

What do you wonder about the code in this program?



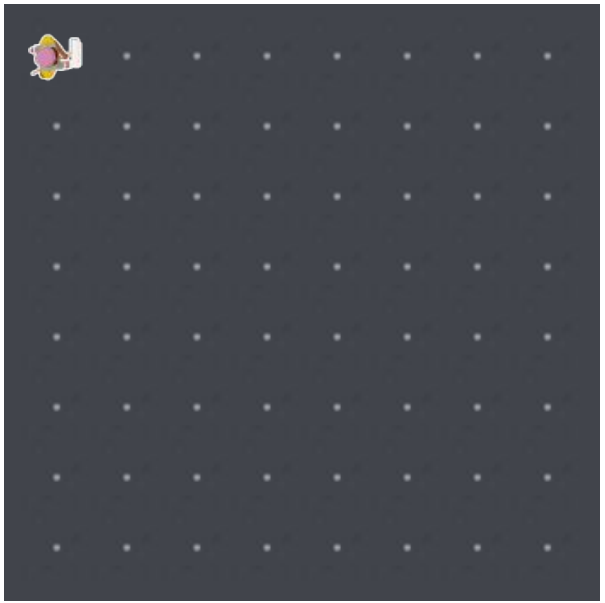
When a constructor is called, it sets the **state** of the object by **assigning the values** to the **instance variables**.



Painter State

```
xLocation = 2
yLocation = 1
direction = "south"
remainingPaint = 0
```

State refers to the **attributes** of an object that are represented by its **instance variables**.



💡 Discuss:

What are some **limitations** of a **no-argument constructor**?

Investigate and Modify

Navigate to Lesson 3, Level 2

✅ Do This:

1. Investigate the code on **Levels 2 through 4**
2. Make changes as prompted and observe the results

What did you discover from the modifications you made to the code?

A **parameterized constructor** is a constructor that has a **specific number of arguments** to be passed to **assign values** to an object's **instance variables**.

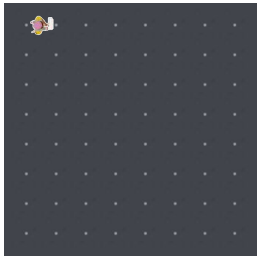
Parameterized Constructor

```
public Painter(int x, int y, String dir, int paint)
```

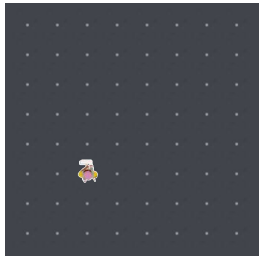
Painter.java

```java
public class Painter {
  private int xLocation;
  private int yLocation;
  private Direction direction;
  private int remainingPaint;

  public Painter() {
    xLocation = 0;
    yLocation = 0;
    direction = Direction.EAST;
    remainingPaint = 0;
  }

  public Painter(int x, int y, String dir, int paint) {
    xLocation = x;
    yLocation = y;
    direction = new Direction(dir);
    remainingPaint = paint;
  }
}
```

Defining **two or more constructors** with the **same name** but with **different signatures** is called **overloading** .



`Painter()`



`Painter(2, 3, "North", 4)`

📝 Unit 2 Guide

🎥 Overloaded Constructors

Why would we want to have multiple constructors?

Complete the guided notes on the 📝 **Unit 2 Guide**.

Retrieve
your knowledge and ideas and write it down silently

Pair
up with a neighbor and talk about your reflections

Share
your thoughts in a class discussion

💡 Discuss:

What are the **similarities** and **differences** between a **no-argument constructor** and a **parameterized constructor**?

📝 **Unit 2 Guide**

💡 Discuss:

- When do you think it's best to use a **no-argument constructor**?
- What about a **parameterized constructor**?

```
public Painter()
public Painter(int x, int y, String dir, int paint)
```

Components of a Parameterized Constructor

```
public Painter(int x, int y, String dir, int paint)
```

the type and name of the values expected when calling the constructor

A formal parameter is the **value to be passed** to a constructor or method.

Components of a Parameterized Constructor

```
public Painter(int x, int y, String dir, int paint) {
  xLocation = x;
  yLocation = y;
  direction = new Direction(dir);
  remainingPaint = paint;
}
```

These are **local variables** , which are variables declared and accessible **within a specific block of code**.

Calling a Parameterized Constructor

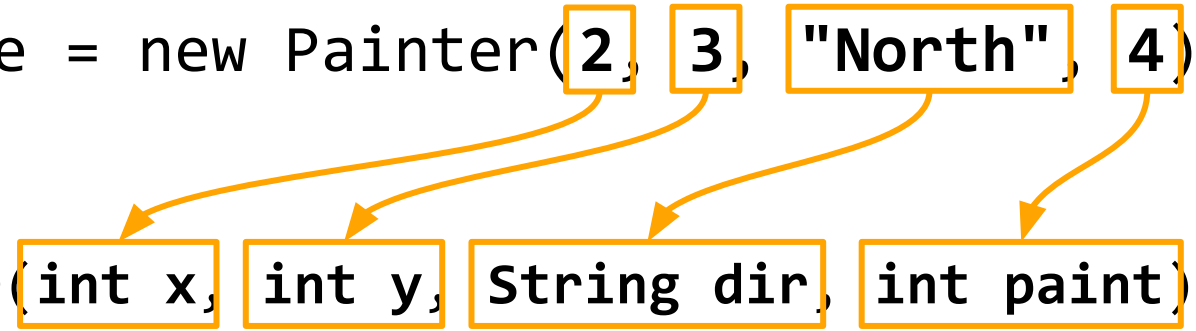
```
Painter katie = new Painter(2, 3, "North", 4);
```

the specific values to assign to the instance variables

An **actual parameter** is the **value to assign** to the formal parameter.

Calling a Parameterized Constructor

```
Painter katie = new Painter(2, 3, "North", 4);

public Painter(int x, int y, String dir, int paint)
```



The values of the **actual parameters** are copied to the constructor's **formal parameters**. This is known as call by value .

🎉 Practice

Navigate to Lesson 3, Level 5

✅ Do This:

1. **Level 5 -** Check for Understanding
2. **Level 6 -** Write a parameterized constructor in a class
3. **Level 7 -** Write multiple parameterized constructors in a class

Wrap Up

Three W's

1. What did we learn today?
2. So what?
3. Now what?

🎉 Today, you learned about . . .

- Writing a parameterized constructor to specify formal parameters
- How to identify the correct constructor being called based on the constructor signature

🚀 Question of the Day

How can I create objects with specific values for its instance variables?

💼 Key Vocabulary (Part 1)

- **state:** the attributes of an object that are represented by its instance variables
- **parameterized constructor:** a constructor that has a specific number of arguments to be passed to assign values to an object's instance variables
- **overloading:** defining two or more constructors or methods with the same name but different signatures

💼 Key Vocabulary (Part 2)

- **formal parameter:** the value to be passed to a constructor or method
- **local variable:** a variable declared and accessible within a specific block of code
- **actual parameter:** the value to assign to the formal parameter
- **call by value:** copying the value of the actual parameter to the constructor's formal parameter