

Lesson 3: Parameterized Constructors

Revisiting the Painter Class

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

creating a `Painter` object at a specific location with a starting amount of paint and facing a specified direction

Writing and Using Constructors

Term	Definition	Example / Picture / Code
state	the attributes of an object that are represented by its instance variables	<pre>Painter State xLocation = 2 yLocation = 1 direction = "south" remainingPaint = 0</pre>
parameterized constructor	a constructor that has a specific number of arguments to be passed to assign values to an object's instance variables	<pre>public Painter(int x, int y, String dir, int paint)</pre>
overloading	defining two or more constructors or methods with the same name but different signatures	<pre>public Painter() public Painter(int x, int y, String dir, int paint)</pre>

Video Notes: Overloaded Constructors

Three Things You Learned

- 1 Parameterized constructor has set number of arguments to be passed
- 2 Parameterized constructor assign user-specified values to instance variables
- 3 Overloading with multiple constructors

Two Interesting Ideas

- 1 Can also overload methods
- 2 Can decide how many constructors to have

One Question or Wonder

Why would we overload methods?

Writing and Using Constructors

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

Both constructors set the starting values for the instance variables when an object is instantiated. These are different because the no-argument constructor sets default values for the instance variables, while the parameterized constructor sets user-specified values.

Annotate the code segment below to identify each component of a parameterized constructor.

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

Term	Definition	Example / Picture / Code
formal parameter	the value to be passed to a constructor or method	<pre>int x in public Painter(int x, int y, String dir, int paint)</pre>
local variable	a variable declared and accessible within a specific block of code	<pre>x in public Painter(int x, int y, String dir, int paint) { xLocation = x; yLocation = y; direction = new Direction(dir); remainingPaint = paint; }</pre>
actual parameter	the value to assign to the formal parameter	<pre>2 in Painter katie = new Painter(2, 3, "North", 4);</pre>
call by value	copying the value of the actual parameter to the constructor's formal parameter	<pre>Painter katie = new Painter(2, 3, "North", 4); public Painter(int x, int y, String dir, int paint)</pre>

Reflection

Question of the Day: How can I create objects with specific values for its instance variables?

Use parameterized constructor

Complete this sentence: I used to think _____, but now I know _____.

Answers may vary.