



Coding-bootcamps.com

Introduction to Java Programming Language

By Jim Sullivan
from [Coding Bootcamps](#)



Introduction to Classes and Objects

Session 3



Brief Recap

- Re-hash what we've covered in our first class

Objective - Discuss The Following

- Classes, Objects and Methods
- Object Instances
- Declaring and Instantiating a Java Object
- Declaring Methods
- set and get Methods
- Initiating Objects with Constructors
- Primitive Types vs. Reference Types

Materials

- These Powerpoint Slides

Object Oriented Programming

- What is an object?
 - A Unique entity that has methods, attributes and can react to events
- What is a method?
 - Things an object can do; the VERB of the object. An action, like show; add; subtract; print.

Object Oriented Programming

- What is an attribute?
 - Things that describe an object; the 'adjective' of an object. In an object this is something like: color; size; enabled
- What is an event?
 - Forces external events to an object to which the object can react.

Object Oriented Programming

- What is a class?
 - Provides a way to create new object based on a definition. For example, The person class or the car class.
- What is a constructor?
 - Special methods used to create new instances of a class. For instance, a Honda Civic is an instance of the car class.

Classes, Objects and Methods

```
public class Dog {  
    String breed;  
    int age;  
    String color;  
  
    void barking() {  
    }  
  
    void hungry() {  
    }  
  
    void sleeping() {  
    }  
}
```

- What is the class ?
- What are the attributes/objects
- What are the methods

Object Instances

- An object and an instance are the same thing.
- An "instance" refers to a specific object of a specific type
 - For example "an instance of type Foo".
 - But when talking about objects in general I would say "objects" rather than "instances".

```
public Dog fido = new Dog();
```

OOP - Classes and Objects: Recap

- What is a class?
 - A data type that allows programmers to create objects
 - Provides a definition for an object, describing an object's attributes (data) and methods (operations)
- What is an object?
 - An object is an instance of the class
 - You can have as many instances as you want

OOP - Declaring Methods

```
public double calculateAnswer(double wingSpan, int numberOfEngines,  
                             double length, double grossTons) {  
    //do the calculation here  
}
```

OOP - Declaring Methods

- The only required elements of a method declaration are
 - The return type
 - The name
 - A pair of parentheses ()
 - A body between braces, {}.

OOP - Declaring Methods

- More generally, method declarations have six components, in order:
 - Modifiers— such as public, private, and others you will learn about later.
 - The return type—the data type of the value returned by the method, or void if the method does not return a value.
 - The method name
 - The parameter list in parenthesis
 - If there are no parameters, you must use empty parentheses.
 - An exception list—to be discussed later.
 - The code that goes between the { } 's

OOP - Encapsulation

- Encapsulation is one of the four fundamental OOP concepts. The other three are inheritance, polymorphism, and abstraction.
- Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit.
- In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.

OOP - Encapsulation

- Get's and Set's are the actual access points of the instance of this class. Therefore, any class that wants to get this data needs to use the 'getter'
- Likewise, if an attribute was private and no 'getter' was present, this data would be hidden

OOP - Encapsulation

- Example

```
public class EncapTest {  
    private String name;  
    private String idNum;  
    private int age;  
  
    public int getAge() {  
        return age;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getIdNum() {  
        return idNum;  
    }  
  
    public void setAge( int newAge) {  
        age = newAge;  
    }  
  
    public void setName(String newName) {  
        name = newName;  
    }  
  
    public void setIdNum( String newId) {  
        idNum = newId;  
    }  
}
```

OOP - Encapsulation - Example

```
/* File name : RunEncap.java */
public class RunEncap {

    public static void main(String args[]) {
        EncapTest encap = new EncapTest();
        encap.setName("James");
        encap.setAge(20);
        encap.setIdNum("12343ms");

        System.out.print("Name : " + encap.getName() + " Age : " +
            encap.getAge());
    }
}
```

OOP - Initiating Objects with Constructors

- As you know, a class provides the blueprint for objects; you create an object from a class.
- Each of the following statements taken from the CreateObjectDemo program creates an object and assigns it to a variable:

```
Point originOne = new Point(23, 94);  
Rectangle rectOne = new Rectangle(originOne, 100, 200);  
Rectangle rectTwo = new Rectangle(50, 100);
```

- The first line creates an object of the Point class, and the second and third lines each create an object of the Rectangle class.

OOP - Initiating Objects with Constructors

- Each of these statements has three parts
- Declaration: The code set in bold are all variable declarations that associate a variable name with an object type.
- Instantiation: The new keyword is a Java operator that creates the object.
- Initialization: The new operator is followed by a call to a constructor, which initializes the new object.

OOP - Primitive Types vs. Reference Types

- Last class we discussed primitive types
 - Who remembers what they were
 - Examples?

OOP - Primitive Types vs. Reference Types

- Reference types are any instantiable class as well as arrays
 - String,
 - Scanner,
 - Random,
 - Die,
 - int[],
 - String[],
 - ArrayList<>
 - HashMap<>

Assignment

- Create an application that uses gets and sets to store some information for a student
 - Think of attributes a student would have

Summary

Live private coaching sessions for Java

- [Private tutoring sessions for software design and engineering- Weekly and monthly plans](#)
- [Java programming language- Private tutoring sessions](#)



Coding-bootcamps.com

Thank You



Coding
Bootcamps