Objects and Classes

Edited by: Huda Saadeh

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

OO Programming Concepts

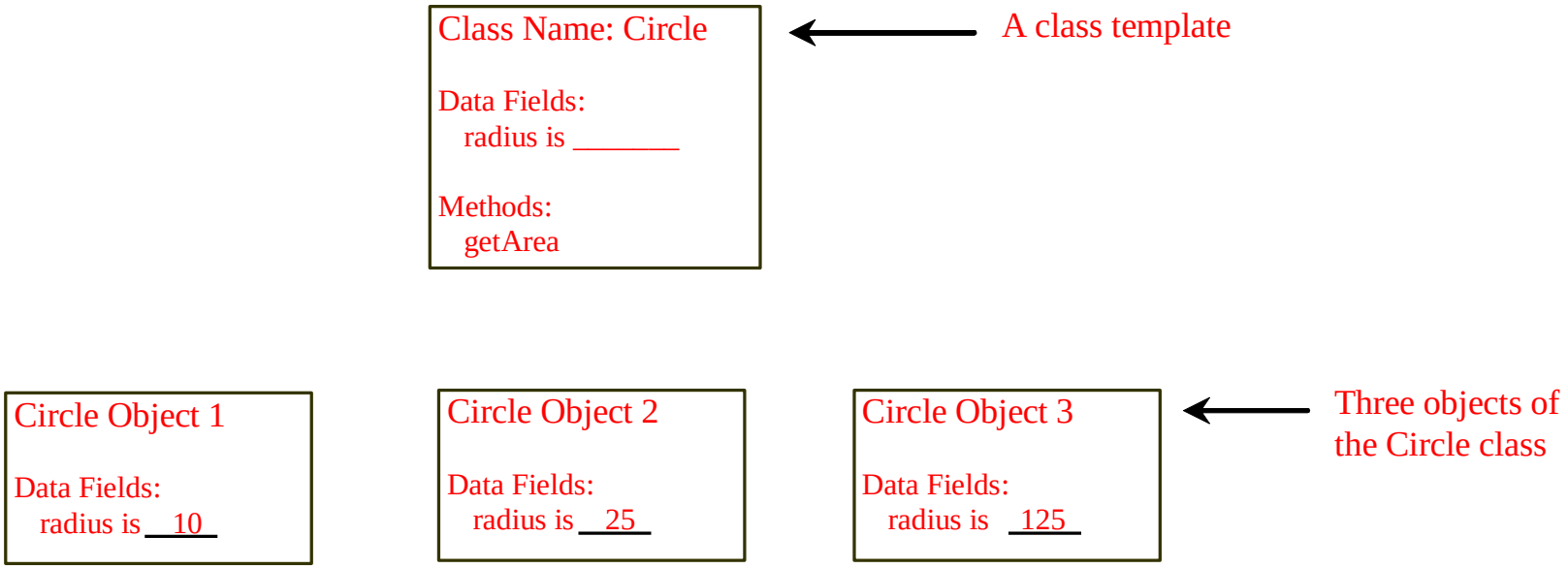
Object-oriented programming (OOP) involves programming using objects. An *object* represents an entity in the real world that can be distinctly identified. For example, a student, a desk, a circle, a button, and even a loan can all be viewed as objects. An object has a unique identity, state, and behaviors. The *state* of an object consists of a set of *data fields* (also known as *properties*) with their current values. The *behavior* of an object is defined by a set of methods or actions.

See the movie

http://www.youtube.com/watch?v=6lvCweI5-x8

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

OBJECTS



An object has both a state and behavior. The state defines the object, and the behavior defines what the object does.

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

CLASSES

Classes are constructs that define objects of the same type.

A Java class uses variables to define data fields and methods to define behaviors.

Additionally, a class provides a special type of methods, known as constructors, which are invoked to construct objects from the class.

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

CLASSES

```
class Circle {
  /** The radius of this circle */
  double radius = 1.0;

  /** Construct a circle object */
  Circle() {
  }

  /** Construct a circle object */
  Circle(double newRadius) {
    radius = newRadius;
  }

  /** Return the area of this circle */
  double getArea() {
    return radius * radius * 3.14159;
  }
}
```

Data field

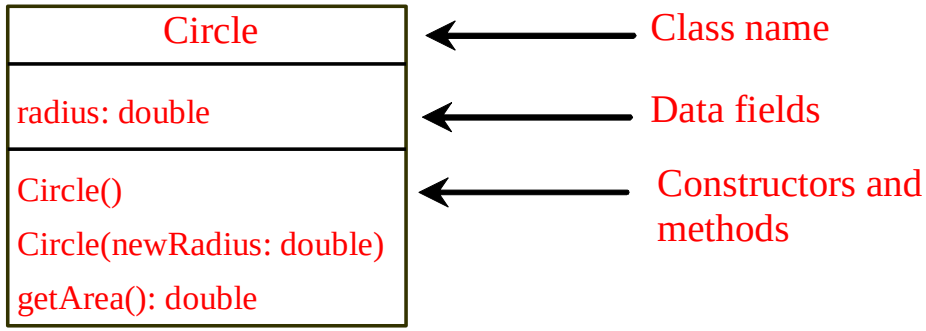
Constructors

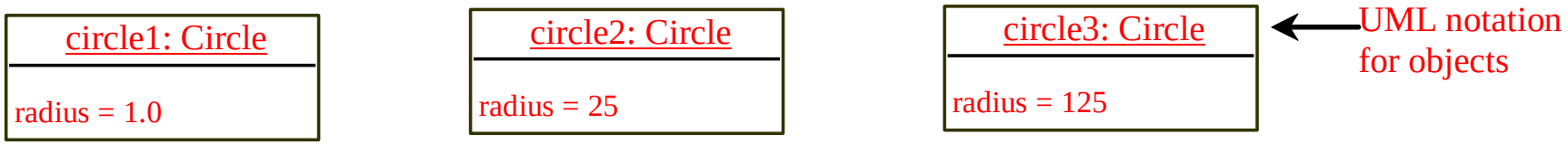
Method

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

UML Class Diagram

UML Class Diagram





Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

EXAMPLE: CREATING OBJECTS

```
public class TestSimpleCircle {
  public static void main(String[] args) {
    // Create a circle with radius 1
    SimpleCircle circle1 = new SimpleCircle();
     System.out.println("The area of the circle of radius " + circle1.radius + " is " +circle1.getArea());

    // Create a circle with radius 25
    SimpleCircle circle2 = new SimpleCircle(25);
    System.out.println("The area of the circle of radius " + circle2.radius + " is " + circle2.getArea());

    // Create a circle with radius 125
    SimpleCircle circle3 = new SimpleCircle(125);
     System.out.println("The area of the circle of radius " + circle3.radius + " is " + circle3.getArea());

    // Modify circle radius
     circle2.radius = 100;
     // or
      circle2.setRadius(100)
      System.out.println("The area of the circle of radius " + circle2.radius + " is " + circle2.getArea());
}
}
```

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

EXAMPLE: DEFINING CLASSES

```
// Define the circle class with two constructors
class SimpleCircle {
        double radius;

         /** Return the area of this circle */
         double getArea() {
                  return radius * radius * Math.PI;
          }
        /** Return the perimeter of this circle */
        double getPerimeter() { return 2 * radius * Math.PI; }
         /** Set a new radius for this circle */
        void setRadius(double newRadius) {
              radius = newRadius; }
          }
```


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

Creating Objects

```
new ClassName();
```

Example:

```
new Circle();
```


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

Declaring Object Reference Variables

To reference an object, assign the object to a reference variable.

To declare a reference variable, use the syntax:

```
ClassName objectRefVar;
```

Example:

```
Circle myCircle;
```


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

Declaring/Creating Objects in a Single Step

To create an object (instantiate an object) we use the new constructor stmt.

```
new ClassName
Example:
new Circle();
```

This object is called (anonymous) You can not use the object created in the previous stmt unless you referenced the object using a reference variable.

```
ClassName objectRefVar = new ClassName();
Or
ClassName objectRefVar;
objectRefVar =new ClassName();
```

Example:

```
Circle myCircle = new Circle();
Or
Circle myCircle;
myCircle= new Circle();
```

Note: The object is also called instance because it is an instance of the class created from.

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

Accessing Object's Members

- Referencing the object's data:

```
objectRefVar.data
```

e.g., `myCircle.radius`

- Invoking the object's method:

```
objectRefVar.methodName(arguments)
```

e.g., `myCircle.getArea()`

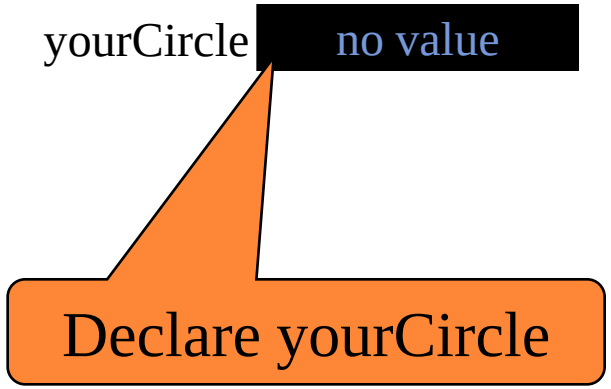
`(take care if the method returns data)`

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

Trace Code, cont.

```
Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

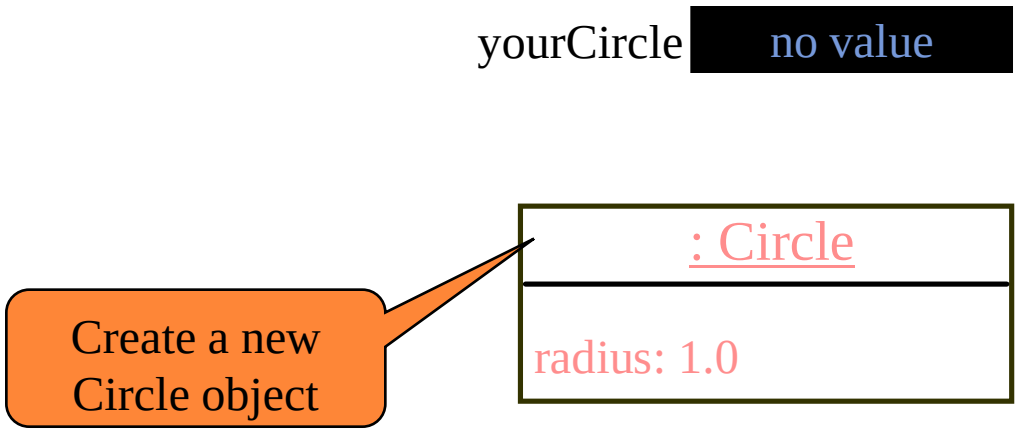


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

TRACE CODE, CONT.

```
Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

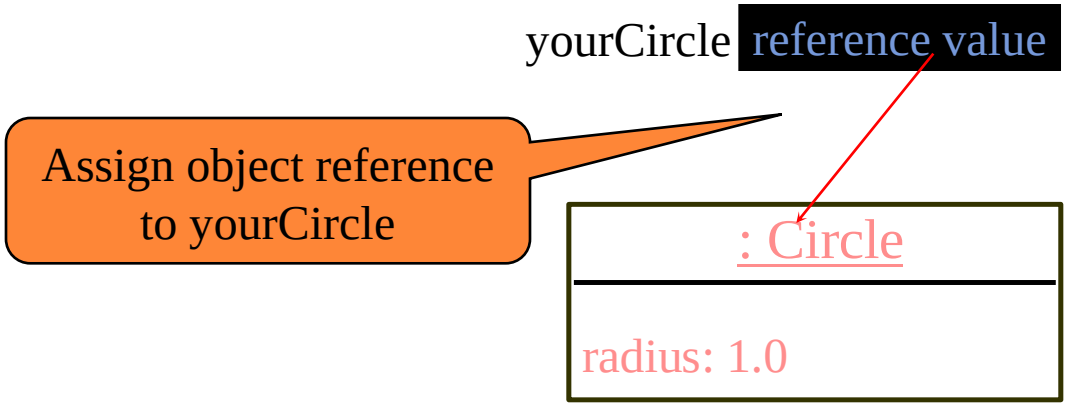


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

TRACE CODE, CONT.

```
Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

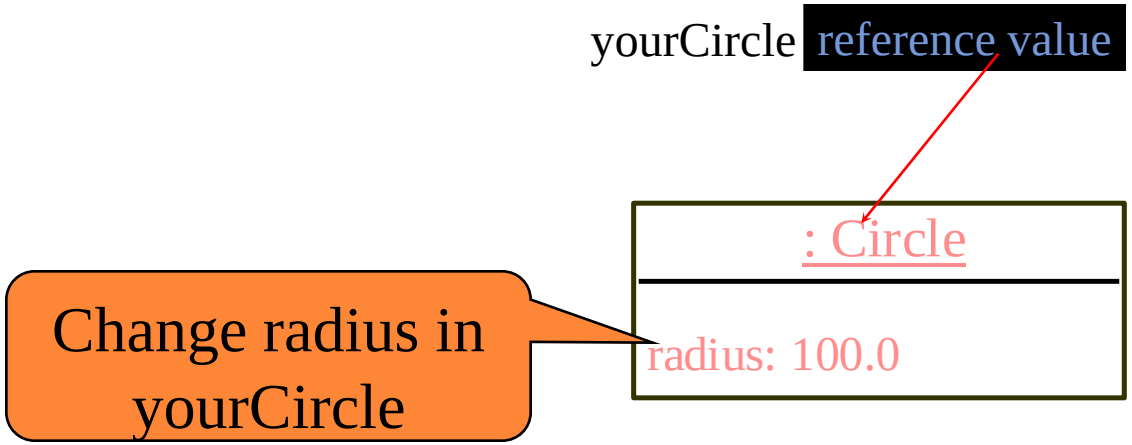


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

TRACE CODE, CONT.

```
Circle yourCircle = new Circle();
```

```
yourCircle.radius = 100;
```



Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

Data Fields

The data fields can be of reference types. For example, the following Student class contains a data field name of the String type.

If a data field of a reference type does not reference any object, the data field holds a special literal value, null.

The default value of a data field is null for a reference type, 0 for a numeric type, false for a boolean type, and '\u0000' for a char type.

However, Java assigns no default value to a local variable inside a method.

Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

Default Value for a Data Field

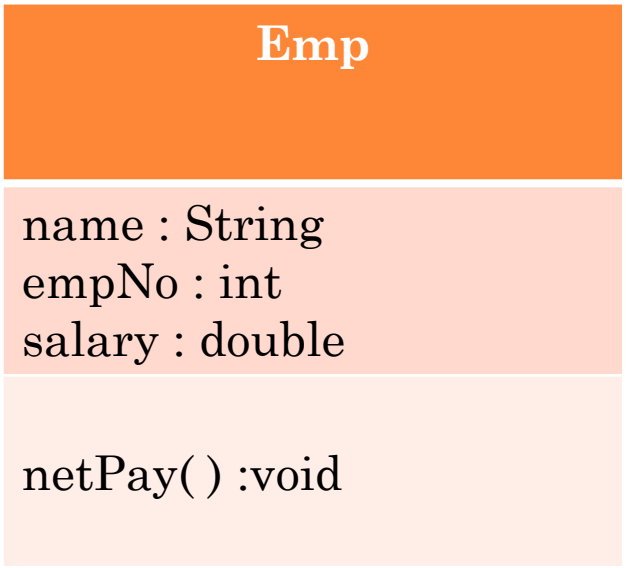
```
public class Student {
  String name; // name has default value null
  int age; // age has default value 0
  boolean isScienceMajor; // isScienceMajor has default value false
  char gender; // c has default value '\u0000'
}

public class Test {
  public static void main(String[] args) {
    int x;  // no default value from java to local variables
    Student student = new Student();
    System.out.println("name? " + student.name);
    System.out.println("age? " + student.age);
    System.out.println("isScienceMajor? " + student.isScienceMajor);
    System.out.println("gender? " + student.gender);
    System.out.println(“x? " + x);
  }
}
```


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

EXAMPLE

- Write the code for the following Employee class which describes employees with the following information(name, empNO, and salary). For each employee the class should calculate his net payment which is his salary after subtracting income tax=0.02



Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

EMP CLASS

```
public class Emp{

 // This is the datafield part
    String name;
    int empNo;
    double salary;

 //This is the action part
    void netPay(){
      double n=salary-salary*0.02;
      System.out.println(n);
    }
}
```


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.

EMPTEST CLASS

```
public class EmpTest {
  public static void main(String[] args) {
    Emp e = new Emp();
    e.name="ahmad";
    e.empNo=1;
    e.salary=100;
    System.out.println("name: " + e.name);
    System.out.println("employee Number: " + e.empNo);
    System.out.println("Salary: " + e.salary);
    System.out.println("Net Payment: " + e.netPay());
  }
}
```


Liang, Introduction to Java Programming, Ninth Edition, (c) 2013 Pearson Education, Inc. All rights reserved.