Introduction to Java for C++ Programmers

Segment - 3

JAC 444

Professor: Mahboob Ali

Objective

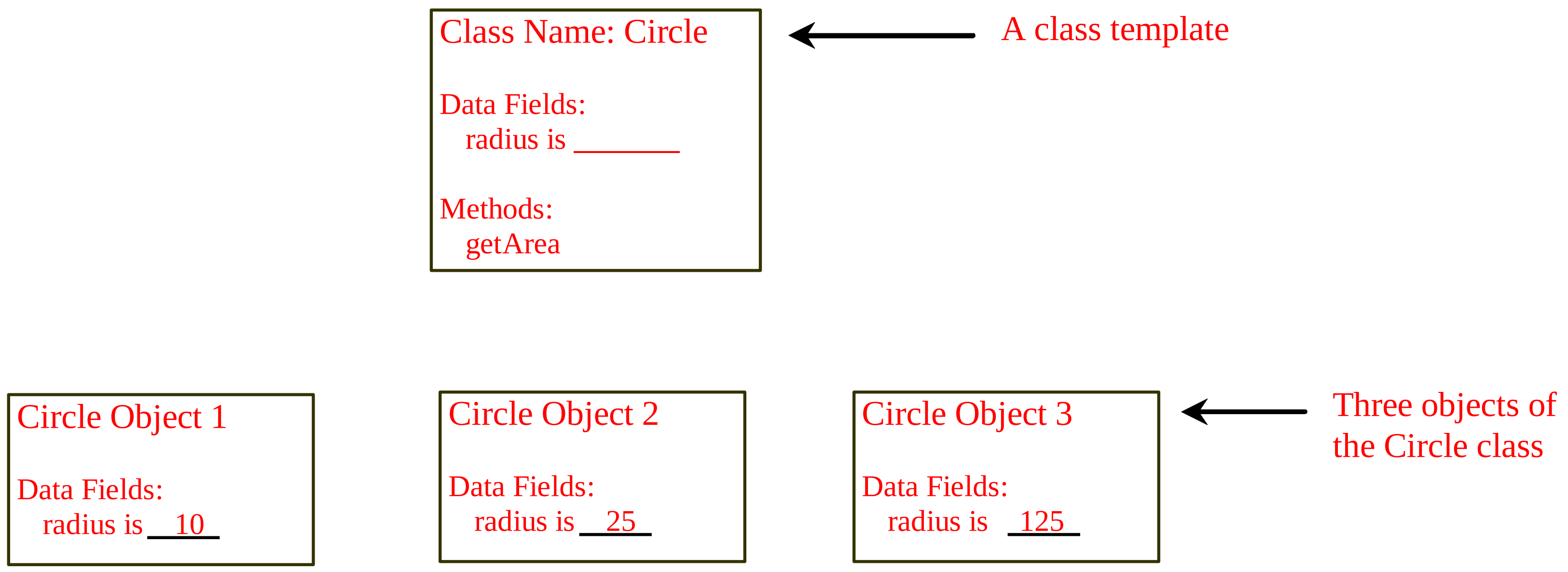
By the end of this segment students should be able to have understanding about:

- Object Oriented programming
- Java Objects
- Java Classes

Object Oriented Programming

• An *object:*	represents an entity in the real world that can be distinctly identified.
• 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.

Objects



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

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.

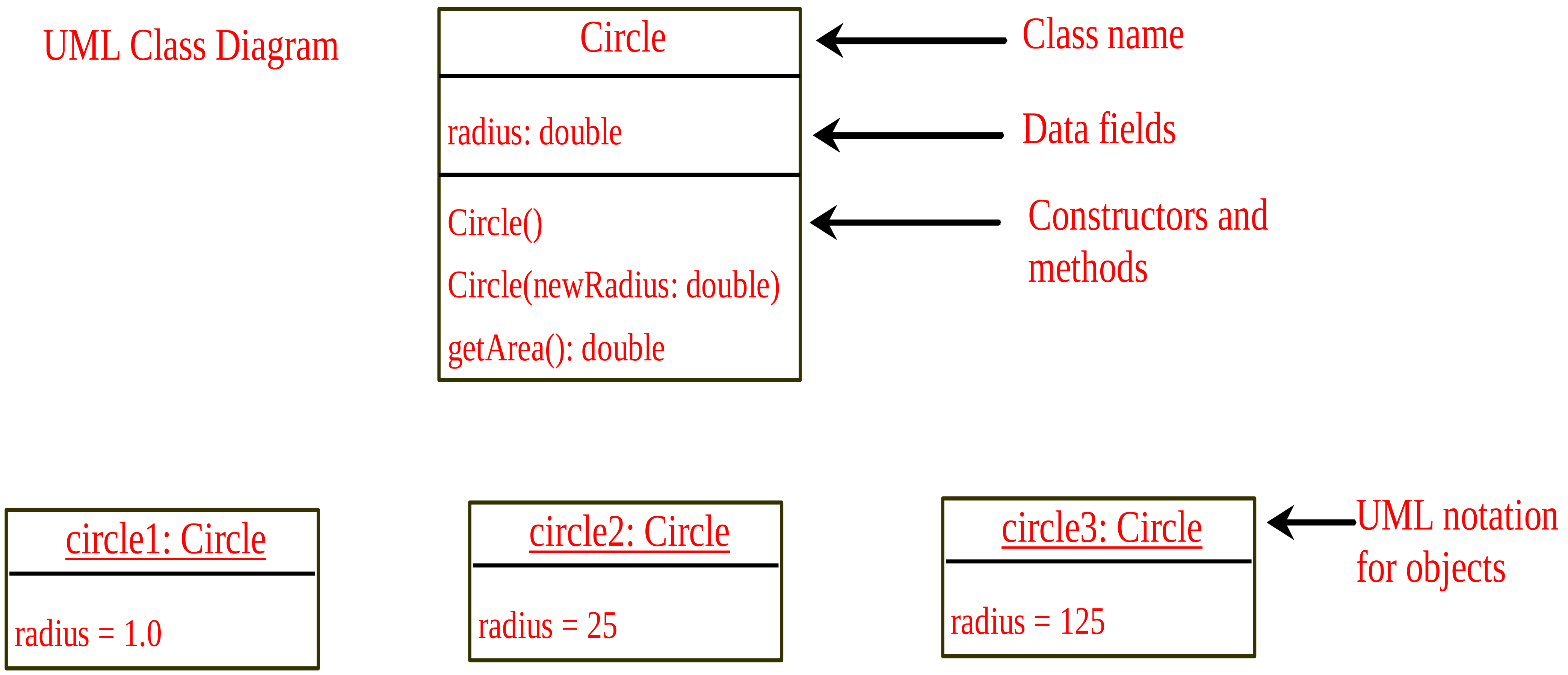
Class Declaration

- The class declaration in Java has the following format:
  - **class ClassName { field(s)**
    constructor(s)
  - **method declaration(s) other class**
    declaration(s)
  - **}**
- The class contains all the code you have to write. Your code must be enclosed by curly braces.

- The object's' life cycle is determined by the elements of the class as following:

1. Objects initializations – Constructors
2. Objects states – Fields
3. Class and its objects behaviors – Methods

UML Class Diagram



Constructors

Constructors are a special kind of methods that are invoked to construct objects.

```
Circle() {//No Argument Constructor

}
                    //With Argument

Circle(double newRadius) {
  radius = newRadius;
}
```

Constructors, cont.

- A constructor with no parameters is referred to as a *no-arg constructor*.
- Constructors must have the same name as the class itself.
- Constructors do not have a return type—not even void.
- Constructors are invoked using the new operator when an object is created. Constructors play the role of initializing objects.

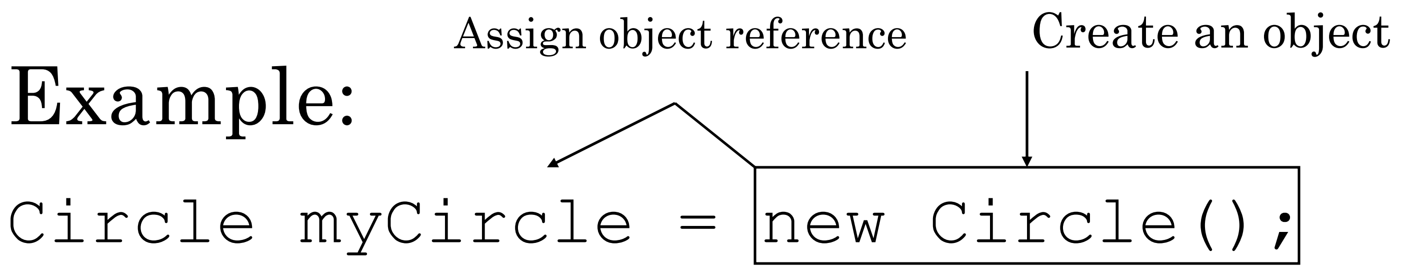
Default Constructor

- A class may be declared without constructors.
- In this case, a no-arg constructor with an empty body is implicitly declared in the class.
- This constructor, called *a default constructor*, is provided automatically *only if no constructors are explicitly declared in the class*.

Declaring/Creating Objects in a Single Step

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

Example:



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

Accessing Objects

- Referencing the object's data:

```
objectRefVar.data
```

e.g., `myCircle.radius`

- Invoking the object's method:

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

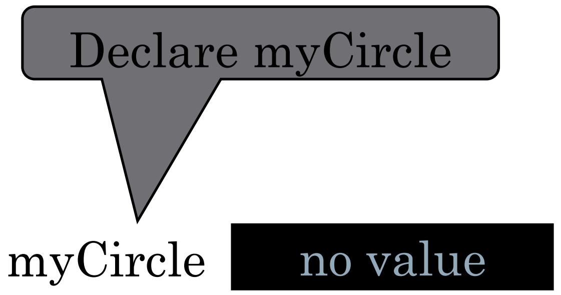
e.g., `myCircle.getArea()`

Trace Code

```
Circle myCircle = new Circle(5.0);

SCircle yourCircle = new Circle();

yourCircle.radius = 100;
```

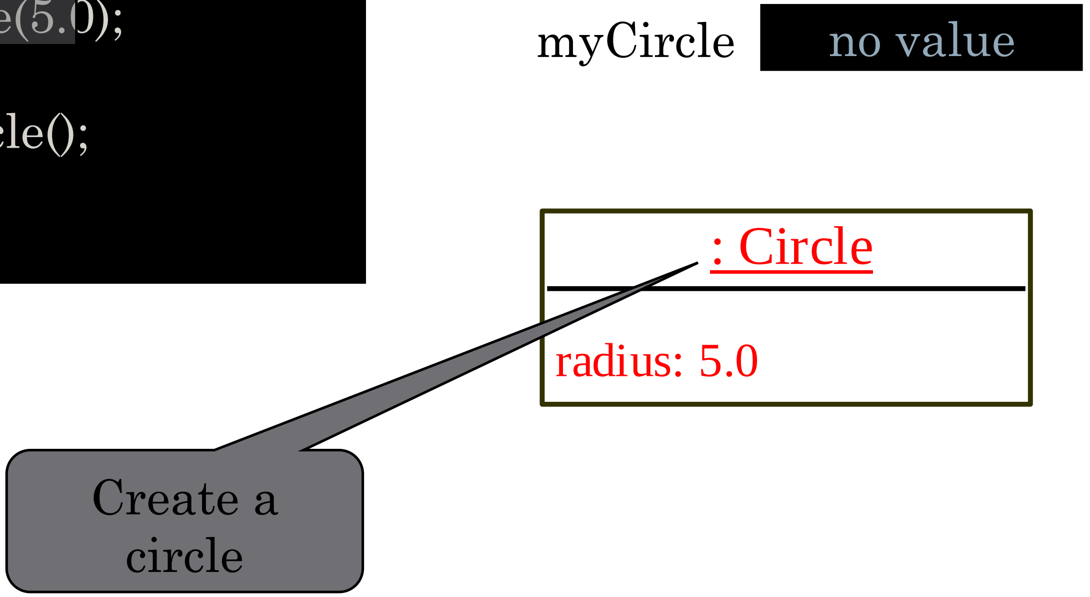


Trace Code, cont.

```
Circle myCircle = new Circle(5.0);

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```



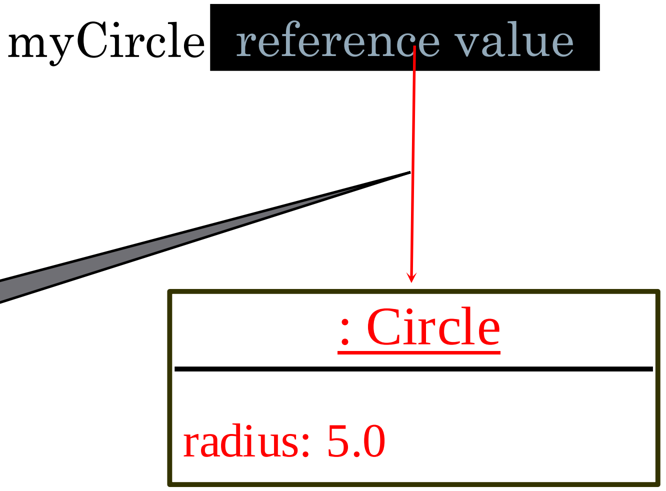
Trace Code, cont.

```
Circle myCircle = new Circle(5.0);

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

Assign object reference to myCircle

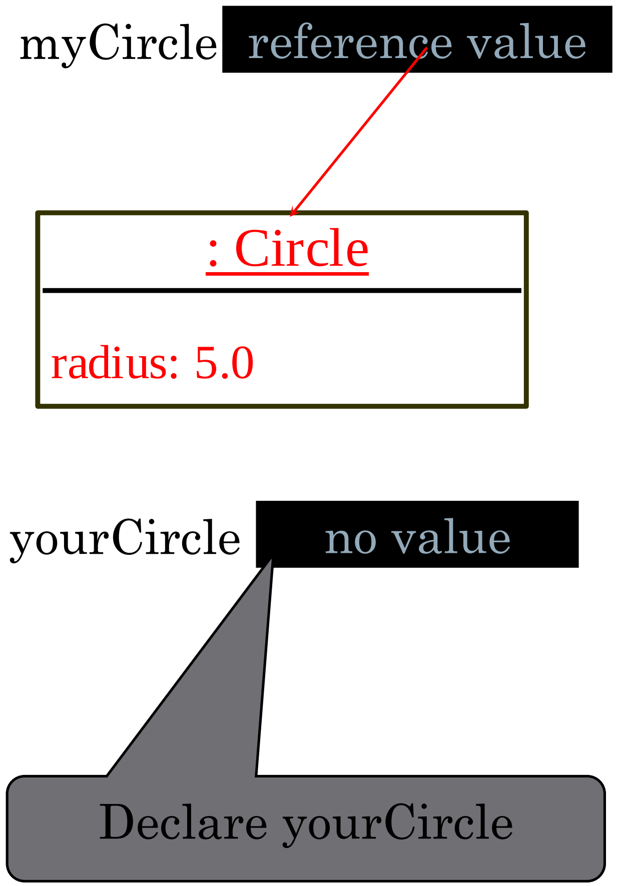


Trace Code, cont.

```
Circle myCircle = new Circle(5.0);

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

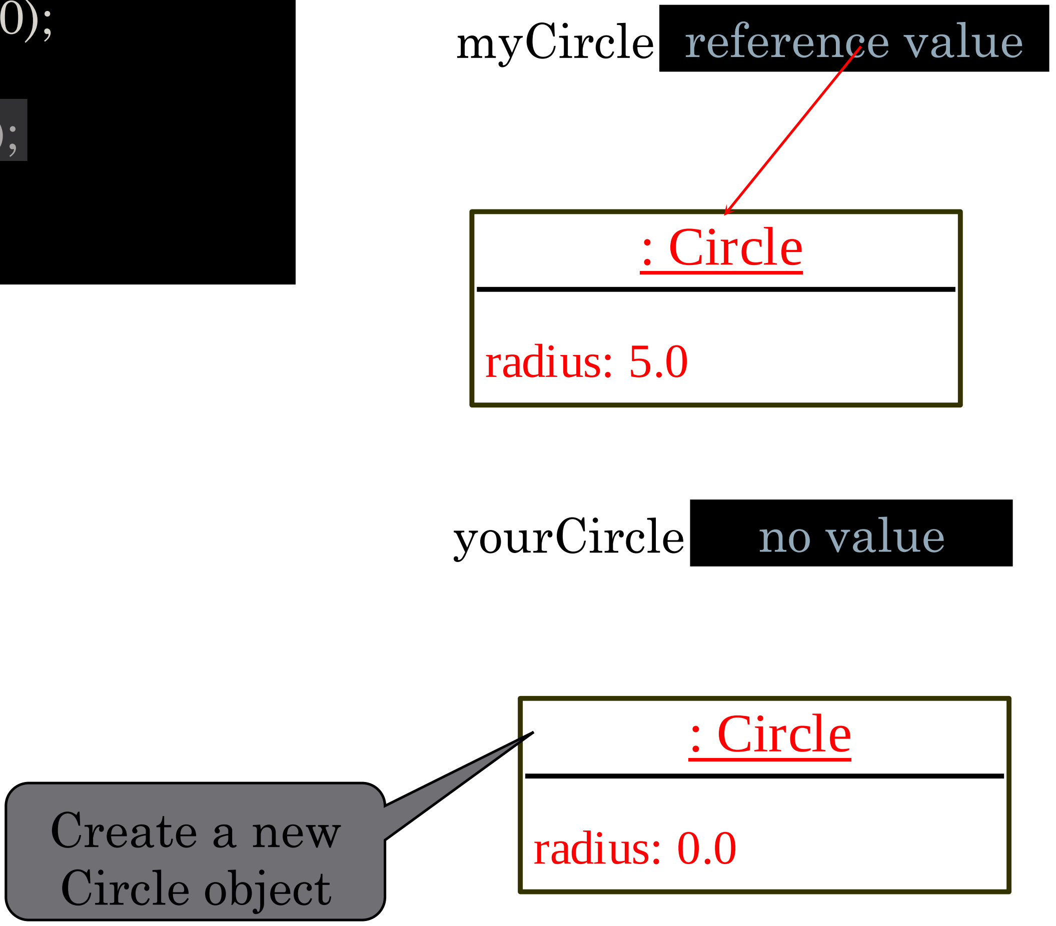


Trace Code, cont.

```
Circle myCircle = new Circle(5.0);

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

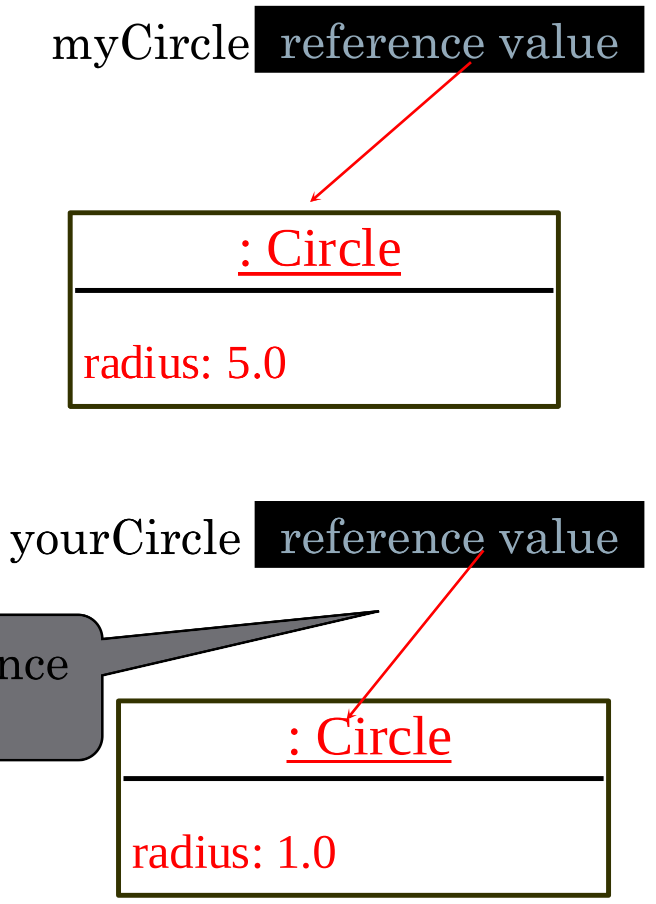


Trace Code, cont.

```
Circle myCircle = new Circle(5.0);

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```



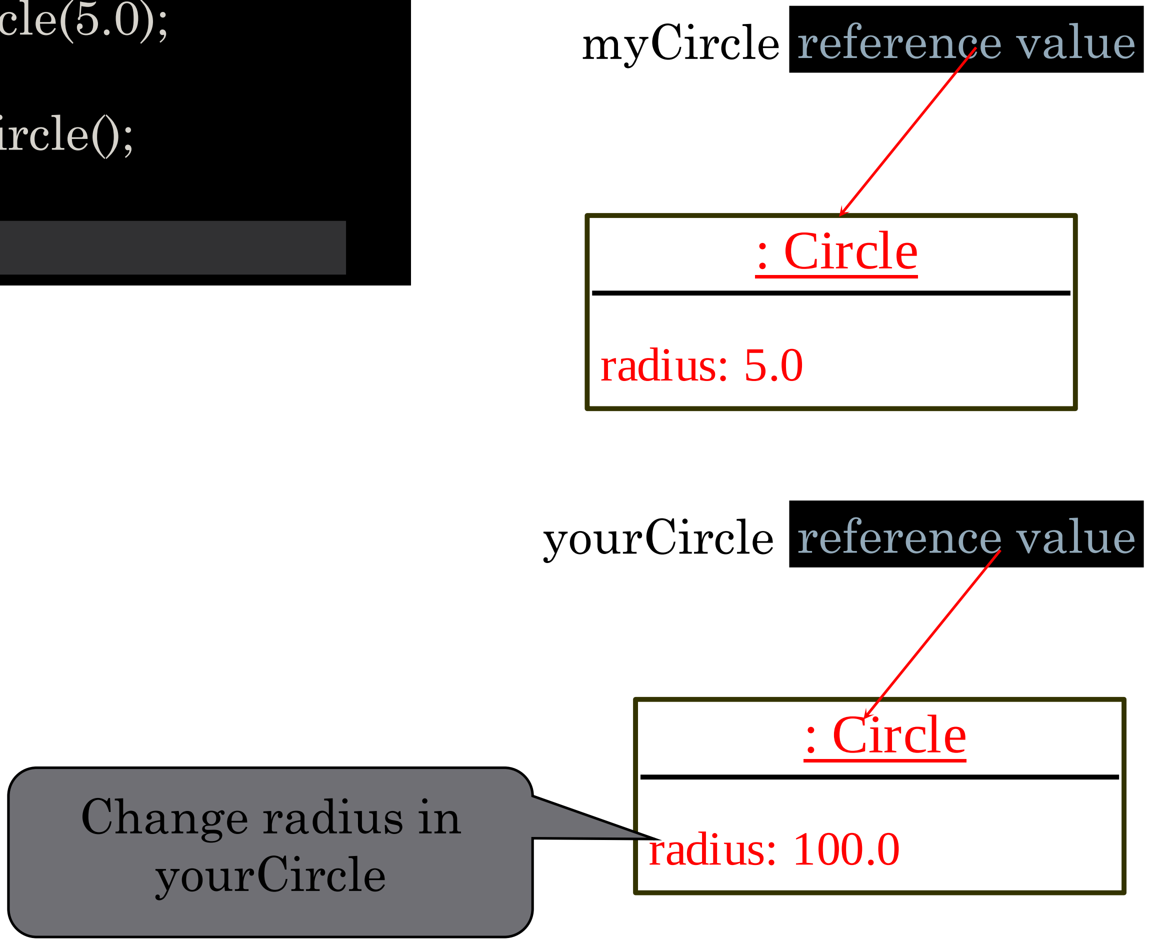
Assign object reference to yourCircle

Trace Code, cont.

```
Circle myCircle = new Circle(5.0);

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```



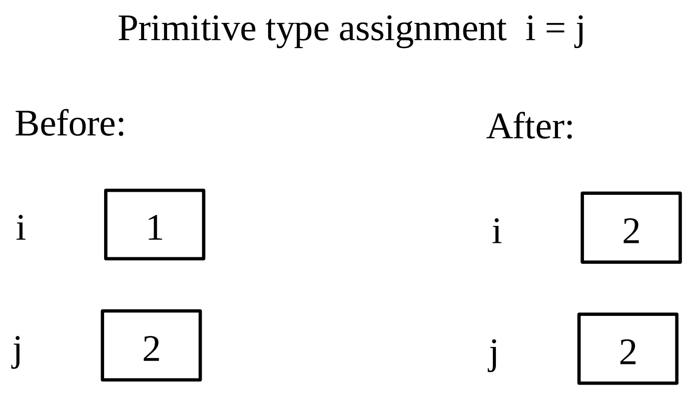
Math.methodName(arguments) e.g., Math.pow(3,2)

Circle1.getArea() ?

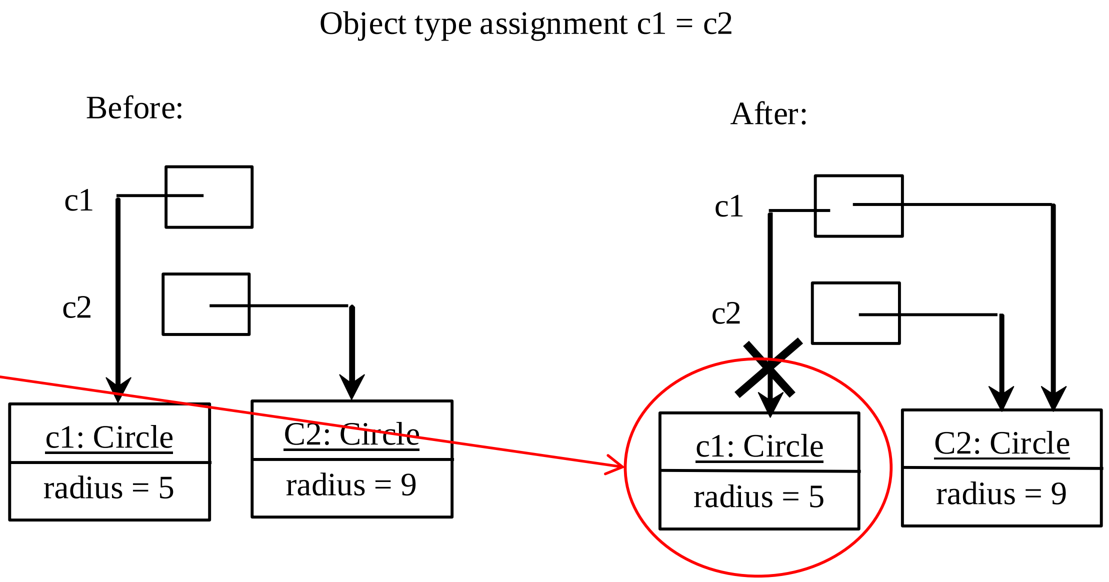
- Methods used in Math class are static methods, which are defined using the static keyword.
- However, getArea() is non-static. It must be invoked from an object using

objectRefVar.methodName(arguments) (e.g., myCircle.getArea()).

Copying Variables of Primitive Data Types and Object Types



Primitive type assignments? i = j
Object type assignments? c1 = c2



Java has automatic Garbage Collection

What will happen to this?