Java Programming AP Edition
U3C9 Objects and Classes

CONSTRUCTORS

ERIC Y. CHOU, PH.D. IEEE SENIOR MEMBER

Constructors

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


```
Circle() {
}

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

Constructors, cont.

A **constructor** with no parameters is referred to as a *no-arg constructor*. If no constructor is given, default one will be used.

- 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.

Creating Objects Using Constructors


```
new ClassName();
```

Example:

```
new Circle();

new Circle(5.0);
```

Declaring Object Reference Variables

To reference an object, assign the object to a reference variable. For an object, you may have as many reference variable (pointer) as you wish.

To declare a reference variable, use the syntax:

```
ClassName objectRefVar;
```

Example:

```
Circle myCircle;
```

Declaring/Creating Objects in a Single Step


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

Example:

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

Accessing Object's Members
(Properties or Methods)

Referencing the object's data:

```
objectRefVar.data
```

e.g., `myCircle.radius`

Invoking the object's method:

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

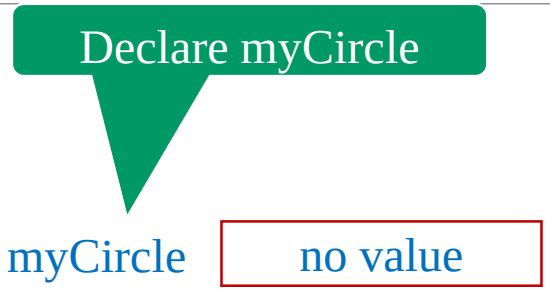
e.g., `myCircle.getArea()`

Trace Code

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

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

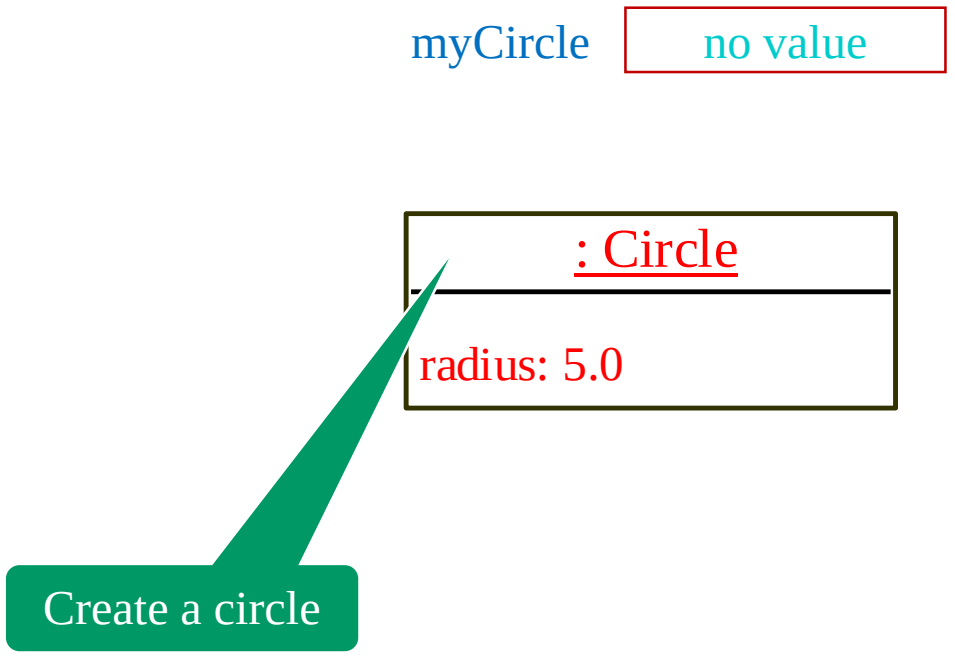


Trace Code

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

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

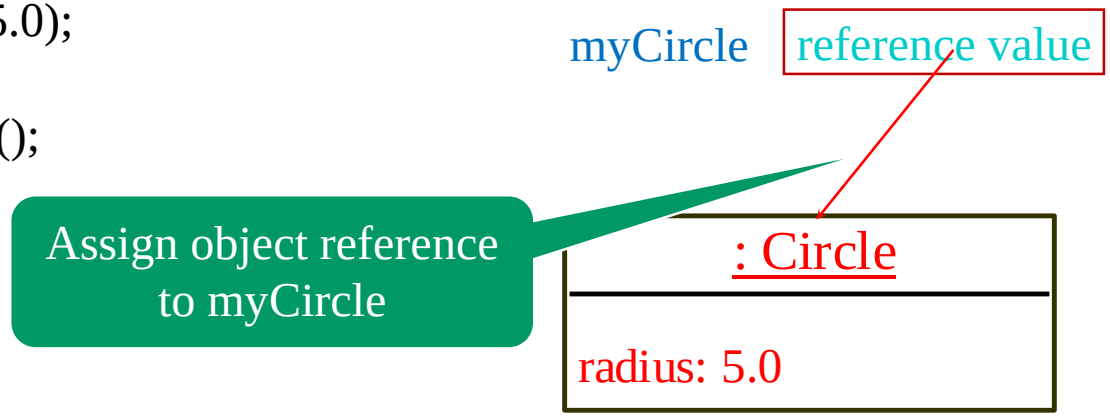


Trace Code

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

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

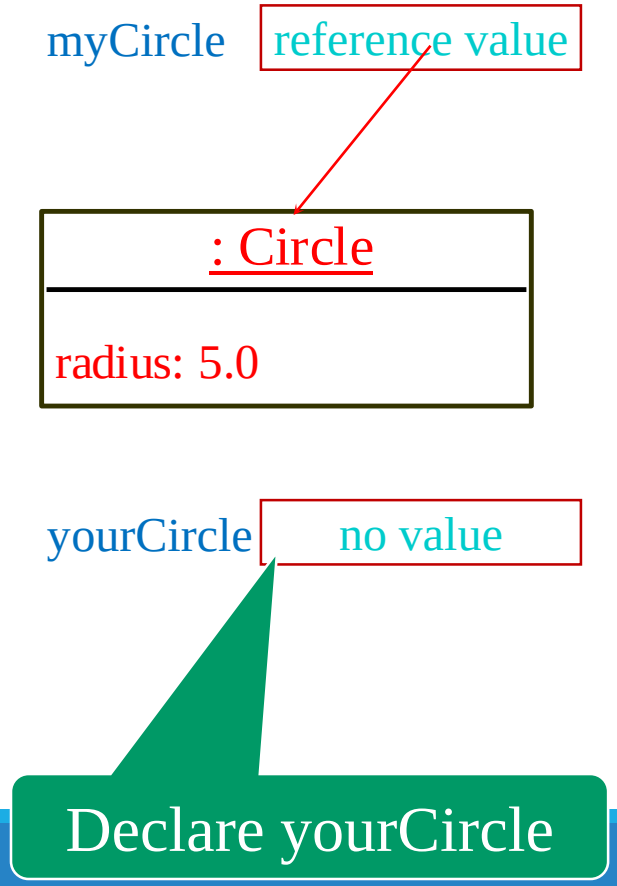


Trace Code

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

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

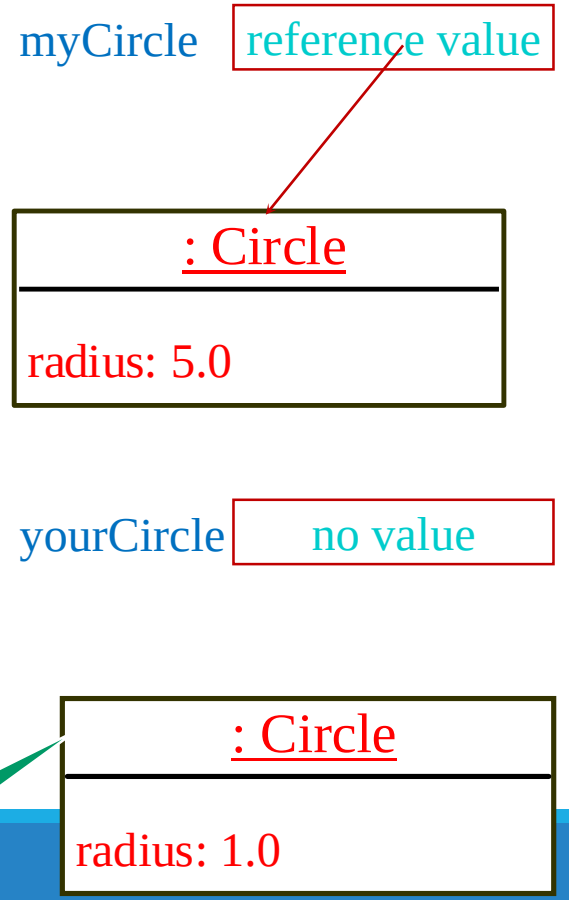


Trace Code

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

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

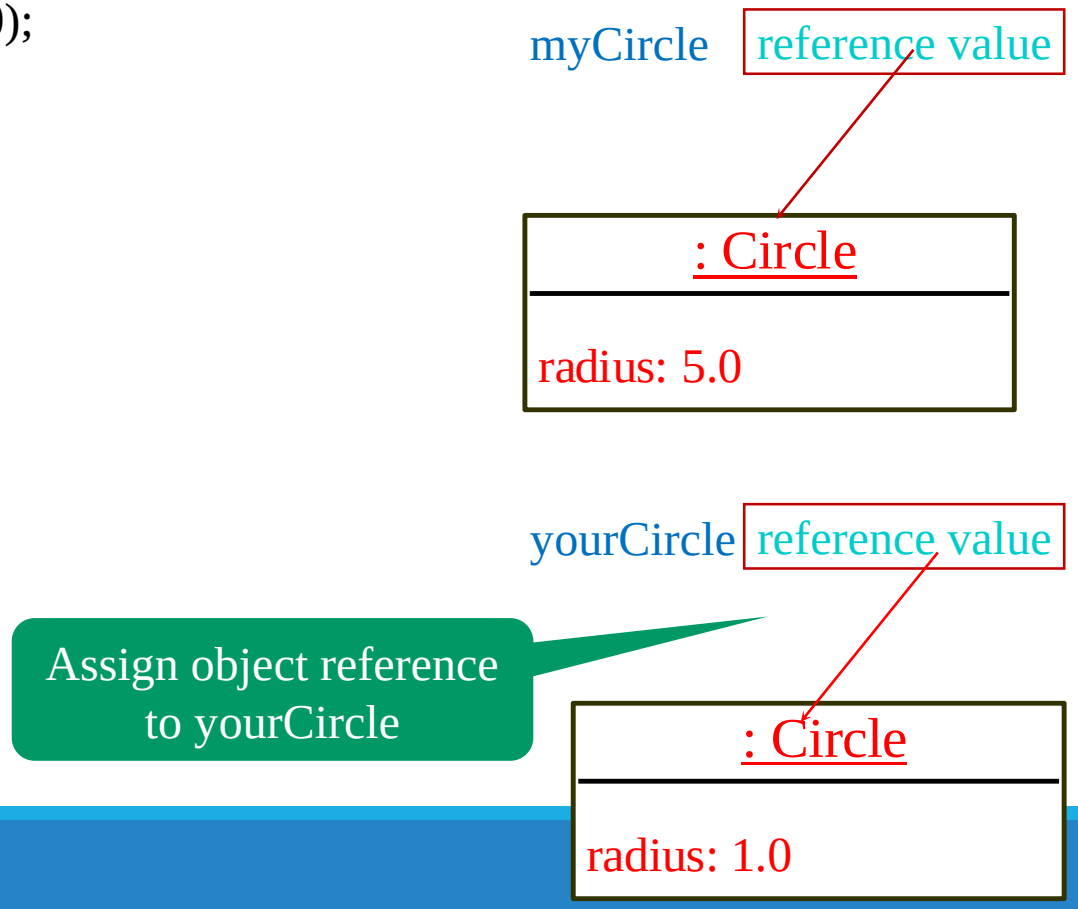


Trace Code

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

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```

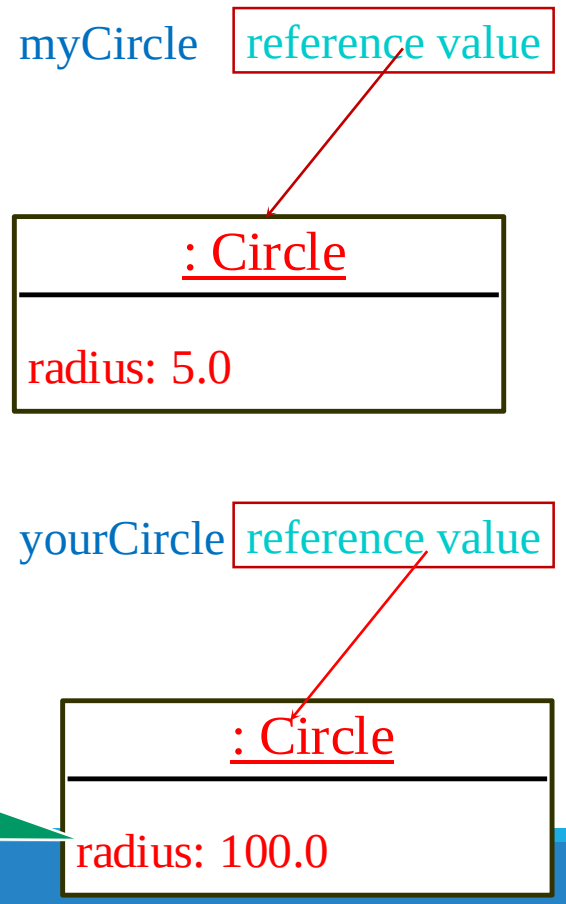


Trace Code

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

Circle yourCircle = new Circle();

yourCircle.radius = 100;
```



Caution

(static members belong to a class, non-static members belongs to objects)

Recall that you use

Math.methodName(arguments) (e.g., Math.pow(3, 2.5))

to invoke a method in the Math class. Can you invoke getArea() using **Circle**.getArea()? The answer is no. All the methods used before this chapter 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()).

More explanations will be given in the section on "Static Variables, Constants, and Methods."