Introducing Classes (Chapter 6 of Schilit)

Object Oriented Programming BS (CS/SE) II

By

Abdul Haseeb

Class Fundamentals

- Core of Java

- Basic foundation of OOP

- Until now you just saw a class which encapsulates main, to demonstrate basics of java

- Class is a template:
  - Defines a data type
  - Create multiple objects using that data type

- Object is an instance of class

General Form of class

```
class classname {
    type instance-variable1;
    type instance-variable2;
    // ...
    type instance-variableN;

    type methodname1(parameter-list) {
      // body of method
    }
    type methodname2(parameter-list) {
      // body of method
    }
    // ...
    type methodnameN(parameter-list) {
        // body of method
    }
}
```

Class Basics

- Contains code and data
- Code and data to gather are called members of a class
- Code organized inside methods
- Variables of class are called Instance variables
  - Because each object will have a different copy of these values

A Simple Class — Code

- Box with only data

```
class Box {
    double width;
    double height;
    double depth;
}
```

A Simple Class — Code

- Two Objects of Box

Use dot operator to access and assign instance variables and methods

How many .class files will be created?

Task

- Create a student datatype having id and name and create two instances of it
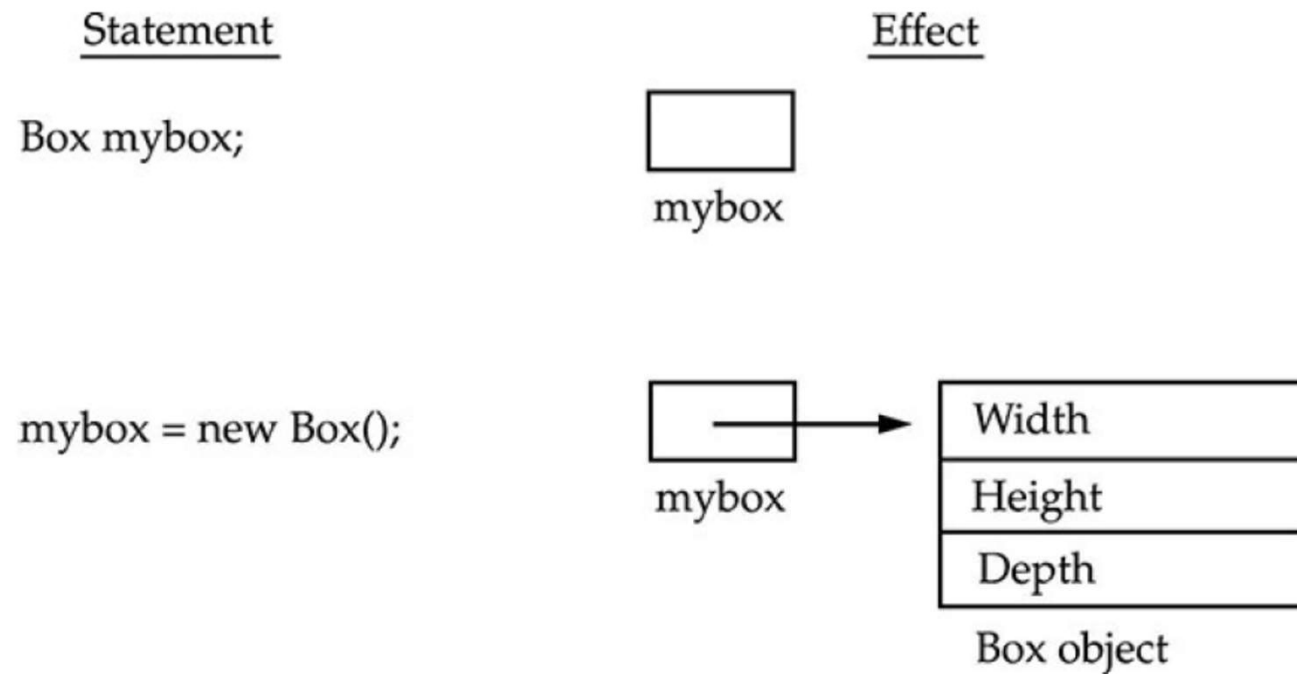
Declaring Objects

Statement | Effect

Box mybox;

mybox

mybox = new Box();

mybox

| Width |
| Height |
| Depth |

Box object

**Figure 6-1** Declaring an object of type **Box**

New Operator

- Creates object of a class, and allocates memory to it
- Performs dynamic allocation (at run time)
- Here classname() refers to constructor
  - Specify what should happen when object is created
  - If no constructor is provided, java provides default constructor

- Both point to same object
- Changes in one affect the others

```
Box b1 = new Box();
Box b2 = b1;
```

Object Reference Variables and use of *null*
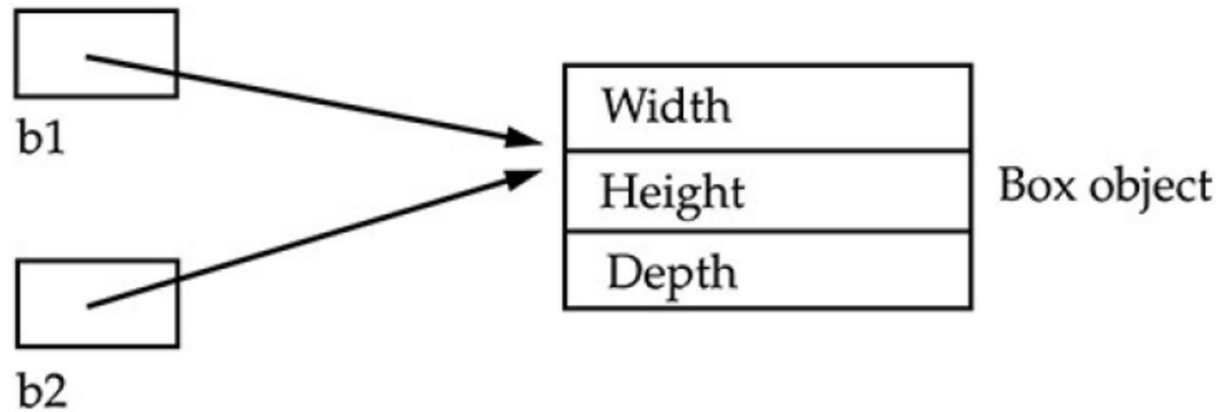
Unhooking through the use of null

```
Box b1 = new Box();
Box b2 = b1;
// ...
b1 = null;
```

Here, **b1** has been set to **null**, but **b2** still points to the original object.

Introducing Methods

```
type name(parameter-list) {
    // body of method
}
```

Inserting the volume method to box class

- Call it with box1
- Call it with box2
- You will notice that when we call instance variables inside a method:
  - No Need to use object reference
- When returning a value:
  - Make sure the return data and method return type are compatible:

Use a method which returns some value