



Java – Overloading Methods & Constructors

B.Sc. 2nd Semester (Paper Code:CC3)

Paulami Basu Ray

Assistant Professor

Department of Computer Science and Applications

Prabhat Kumar College, Contai

A decorative graphic on the left side of the slide. It features a dark blue vertical bar on the far left. A black arrow points to the right from the top of this bar. Several thin, light blue lines curve upwards and to the right from the bottom of the bar, overlapping the text area.

Method Overloading

- In Java, it is possible to define two or more methods in the same class with the same name as long as their parameter declaration are different.
- When this is the case, the methods are said to be overloaded and the process is known as Method Overloading.
- Method overloading is one of the ways that Java supports polymorphism.
- When an overloaded method is invoked, Java uses the type and/or number of arguments as its guide to determine which version of the overloaded method to actually call.

Sample Program

```
Overload.java x
1 //Author: Paulami Basu Ray
2 //file name= Overload.java
3
4 class OverloadDemo{
5     void test(){
6         System.out.println("No parameters");
7     }
8     void test(int a){
9         System.out.println("a= "+a);
10    }
11    void test(int a,int b){
12        System.out.println("a and b: "+a+" "+b);
13    }
14    void test(double a){
15        System.out.println("double a= "+a);
16    }
17 }
18 class Overload{
19     public static void main(String[] args){
20         OverloadDemo ob=new OverloadDemo();
21         ob.test();
22         ob.test(17.39);
23         ob.test(5,8);
24         ob.test(100);
25     }
26 }
```

Output

```
C:\ Command Prompt
D:\P.K College\B.Sc. 2nd Sem\Java Programs>javac Overload.java
D:\P.K College\B.Sc. 2nd Sem\Java Programs>java Overload
No parameters
double a= 17.39
a and b: 5 8
a= 100
D:\P.K College\B.Sc. 2nd Sem\Java Programs>
```

A decorative graphic on the left side of the slide. It features a dark grey arrow pointing to the right at the top. Below the arrow, several thin, curved lines in shades of blue and grey sweep downwards and to the right, creating a sense of movement and design.

Constructors in Java

- ▶ Constructors are used to initialize the object's state. Like methods, a constructor also contains collection of statements(i.e. instructions) that are executed at time of Object creation.
- ▶ **When is a Constructor called ?**
Each time an object is created using **new()** keyword at least one constructor (it could be default constructor) is invoked to assign initial values to the data members of the same class.

Constructor

```
ConstructorDemo.java x
1 //Author: Paulami Basu Ray
2 //file name= Overload.java
3 class Box {
4     double width;
5     double height;
6     double depth;
7     // This is the constructor for Box.
8     Box(double w, double h, double d) {
9         width = w;
10        height = h;
11        depth = d;
12    }
13    // compute and return volume
14    double volume() {
15        return width * height * depth;
16    }
17 }
18 class ConstructorDemo{
19     public static void main(String[] args)
20     {
21         Box ob=new Box(2.1,5.6,7.8);
22         System.out.println("Volume= "+ob.volume());
23     }
24 }
```

Constructor Output

C:\> Command Prompt

```
D:\P.K College\B.Sc. 2nd Sem\Java Programs>jawac ConstructorDemo.java
```

```
D:\P.K College\B.Sc. 2nd Sem\Java Programs>java ConstructorDemo  
Volume= 91.728
```

```
D:\P.K College\B.Sc. 2nd Sem\Java Programs>
```

Constructor Overloading

```
OverloadConstructor.java x
1 //Author: Paulami Basu Ray
2 //file name= OverloadConstructor.java
3 class Box {
4     double width;
5     double height;
6     double depth;
7     // This is the constructor for Box.
8     Box(double w, double h, double d) {
9         width = w;
10        height = h;
11        depth = d;
12    }
13    // constructor used when no dimensions specified
14    Box() {
15        width = -1; // use -1 to indicate
16        height = -1; // an uninitialized
17        depth = 1; // box
18    }
19    // constructor used when cube is created
20    Box(double len) {
21        width = height = depth = len;
22    }
23    // compute and return volume
24    double volume() {
25        return width * height * depth;
26    }
27 }
28 class OverloadConstructor{
29     public static void main(String[] args)
30     {
31         Box ob1=new Box(2.1,5.6,7.8);
32         System.out.println("Volume= "+ob1.volume());
33
34         Box ob2=new Box(2);
35         System.out.println("Volume= "+ob2.volume());
36
37         Box ob3=new Box();
38         System.out.println("Volume= "+ob3.volume());
39     }
40 }
```


Output

Command Prompt

```
D:\P.K College\B.Sc. 2nd Sem\Java Programs>javac OverloadConstructor.java
```

```
D:\P.K College\B.Sc. 2nd Sem\Java Programs>java OverloadConstructor
```

```
Volume= 91.728
```

```
Volume= 8.0
```

```
Volume= -1.0
```

```
D:\P.K College\B.Sc. 2nd Sem\Java Programs>
```