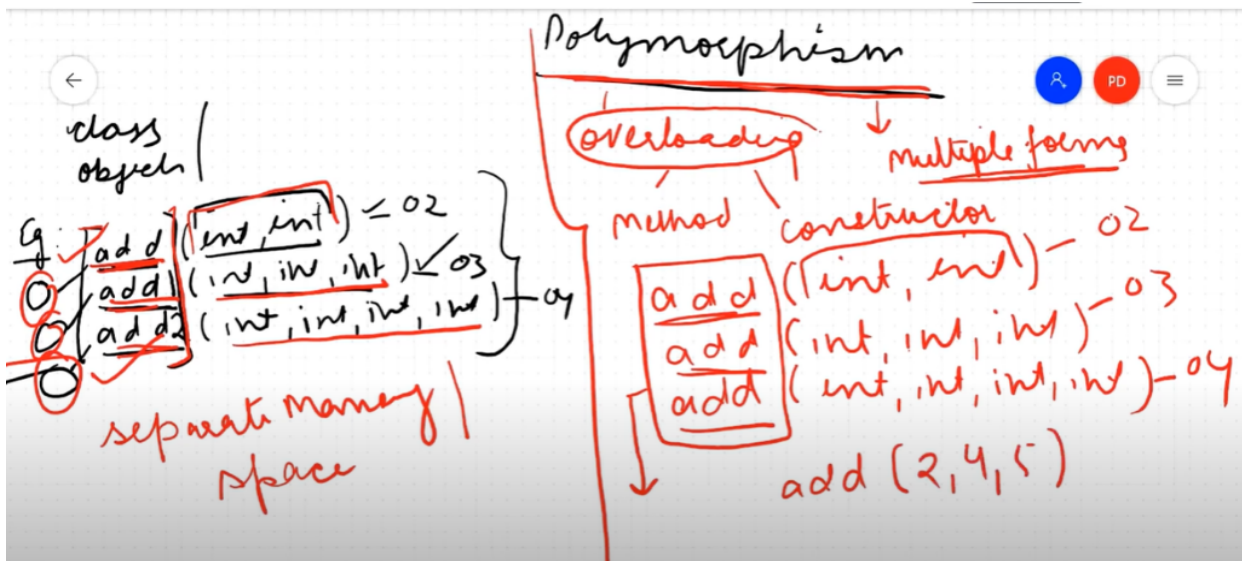


Session-13 Polymorphism

Wednesday, August 24, 2022 6:00 PM



Method Overloading:- is used to save the memory. Because it is pointing to the same method

Method Overloading

With method overloading, multiple methods can have the same name with different parameters

Example

```
int myMethod(int x)
```

```
float myMethod(float x)
```

```
double myMethod(double x, double y)
```

```

1 package java_hub;
2
3 public class New_044_Methodoverloading {
4     // In Method overloading, multiple methods can have
5     // 1. Same Name
6     // 2. Different parameter number
7     // 3. Different parameter data types
8
9     void add(int a, int b) {
10         System.out.println("1:" + (a + b));
11     }
12
13     void add(double a, double b) {
14         System.out.println("2:" + (a + b));
15     }
16
17     void add(int a, double b) {
18         System.out.println("3:" + (a + b));
19     }
20
21     void add(int a, int b, int c) {
22         System.out.println("4:" + (a + b));
23     }
24
25     void add(int a, int b, int c) {
26         System.out.println("4:" + (a + b));
27     }
28
29 }

```

Console: <terminated> New_044_Methodoverloading [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (24-Aug-2022, 6:11:13 pm - 6:11:13 pm) [pid: 23904]

1:5
2:464.572
3:234.34
4:264

```

22      System.out.println("3:" + (a + b));
23    }
24
25    void add(int a, int b, int c) {
26
27      System.out.println("4:" + (a + b));
28    }
29
30  }
31
32  public static void main(String[] args) {
33
34    //Create object
35
36    New_044_Methodoverloading obj1 = new New_044_Methodoverloading();
37    obj1.add(2, 3);
38    obj1.add(232, 232, 232, 34);
39    obj1.add(2, 232, 34);
40    obj1.add(34, 230, 20);
41
42  }
43
44  }
45
46  }
47
48  }
49

```

Console Output:

```

1:5
2:464.572
3:234.34
4:264

```

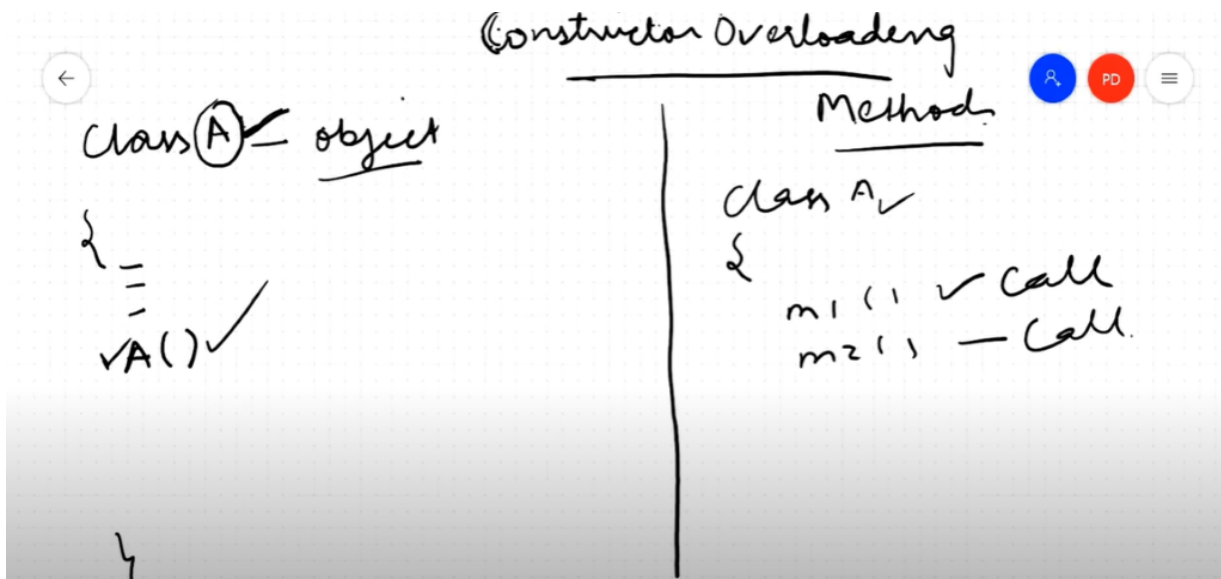
Advantage of method overloading

- Save memory
- *Increases the readability of the program*

Different ways to overload the method

There are two ways to overload the method in java

- By changing number of parameters
- By changing the data type



Constructor –

- The name of the class and the constructor should be same.
- Is called whenever we are creating the object of it.
- Does not have any return type.

```

1 package java_hub;
2
3 public class New_044_ConstructorOverloading {

```

```

4
5 New_044_ConstructorOverloading(int x) {
6     System.out.println(x);
7 }
8
9 New_044_ConstructorOverloading(double x) {
10     System.out.println(x);
11 }
12
13
14
15 New_044_ConstructorOverloading(double x, double y) {
16     System.out.println(x);
17     System.out.println(y);
18 }
19
20
21
22 public static void main(String[] args) {
23
24     New_044_ConstructorOverloading obj1 = new New_044_ConstructorOverloading(2);
25     New_044_ConstructorOverloading obj2 = new New_044_ConstructorOverloading(223.23, 232.23);
26     New_044_ConstructorOverloading obj3 = new New_044_ConstructorOverloading(212321.23, 31232.2);
27
28 }

```

Console × Problems × Debug Shell × Coverage × TestNG
 <terminated> New_044_ConstructorOverloading [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (24-Aug-2022, 6:18:56 pm – 6:18:57 pm) [pid: 24104]
 2
 223.23
 232.23
 212321.23
 31232.2

For_enchance... String1.java String2.java finalkey.java New_042_Sta... New_044_Met... New_044_Con... New_044_Mai... × 1s

```

1 package java_hub;
2
3 public class New_044_Mainmethodoverlaoading {
4
5
6     void main(int x) {
7         System.out.println(x);
8     }
9
10    void main(double x,int y) {
11        System.out.println(x);
12    }
13
14    void main(double x) {
15        System.out.println(x);
16    }
17
18
19    // Entry point
20    public static void main(String[] args) {
21
22        New_044_Mainmethodoverlaoading obj = new New_044_Mainmethodoverlaoading();
23        obj.main(3);
24        obj.main(23.23);
25        obj.main(232.23, 20);
26
27    }
28
29 }
30
31

```

Console × Problems × Debug Shell × Coverage × TestNG
 <terminated> New_044_Mainmethodoverlaoading [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (24-Aug-2022, 6:25:31 pm – 6:25:31 pm) [pid: 18772]
 3
 23.23
 232.23