

NAME – RAJDEEP JAISWAL

DATE – 16 NOV 2021

BRANCH – BTECH CSE

SEC = 608 A

UID -20BCS2761

SUB- JAVA Worksheet

1. Aim/Overview of the practical:

Write a program to differentiate between method overloading and method overriding.

Method Overloading in Java

If a class has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

Code in Text –

```
// method overloading in Java.
package com.company;
public class pra {

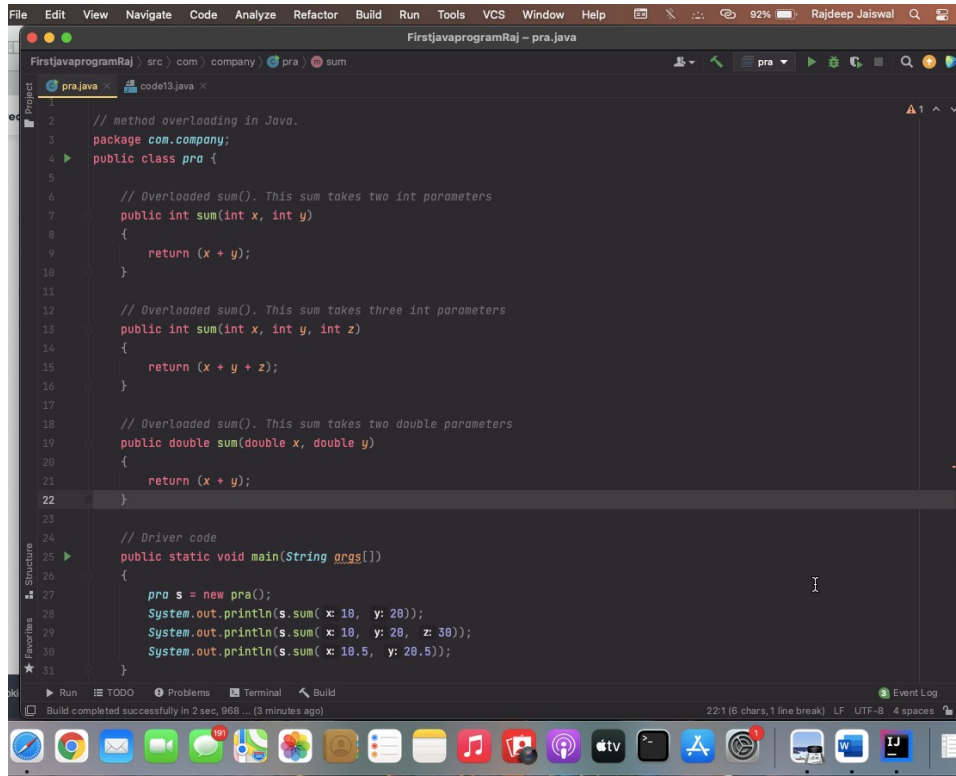
    // Overloaded sum(). This sum takes two int parameters
    public int sum(int x, int y)
    {
        return (x + y);
    }

    // Overloaded sum(). This sum takes three int parameters
    public int sum(int x, int y, int z)
    {
        return (x + y + z);
    }

    // Overloaded sum(). This sum takes two double parameters
    public double sum(double x, double y)
    {
        return (x + y);
    }

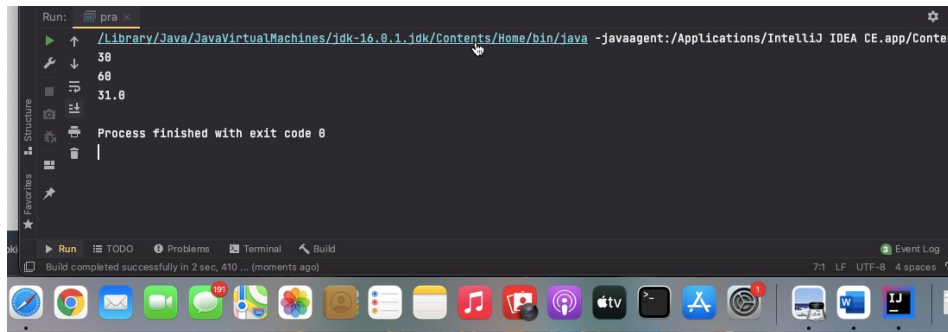
    // Driver code
    public static void main(String args[])
    {
        pra s = new pra();
        System.out.println(s.sum(10, 20));
        System.out.println(s.sum(10, 20, 30));
        System.out.println(s.sum(10.5, 20.5));
    }
}
```

Code in Compiler –



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help 92% Rajdeep Jaiswal
FirstjavaprogramRaj - pra.java
FirstjavaprogramRaj | src | com | company | pra | sum
pra.java | code13.java
1
2 // method overloading in Java.
3 package com.company;
4 public class pra {
5
6     // Overloaded sum(). This sum takes two int parameters
7     public int sum(int x, int y)
8     {
9         return (x + y);
10    }
11
12    // Overloaded sum(). This sum takes three int parameters
13    public int sum(int x, int y, int z)
14    {
15        return (x + y + z);
16    }
17
18    // Overloaded sum(). This sum takes two double parameters
19    public double sum(double x, double y)
20    {
21        return (x + y);
22    }
23
24    // Driver code
25    public static void main(String args[])
26    {
27        pra s = new pra();
28        System.out.println(s.sum(x: 10, y: 20));
29        System.out.println(s.sum(x: 10, y: 20, z: 30));
30        System.out.println(s.sum(x: 10.5, y: 20.5));
31    }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
Run TODO Problems Terminal Build Event Log
Build completed successfully in 2 sec, 968 ... (3 minutes ago) 22:1 (6 chars, 1 line break) LF UTF-8 4 spaces
```

OUTPUT –



```
Run: pra x
/Library/Java/JavaVirtualMachines/jdk-16.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Content
30
38
68
31.0
Process finished with exit code 0
Build completed successfully in 2 sec, 410 ... (moments ago) 7:1 LF UTF-8 4 spaces
```

Method Overriding in Java

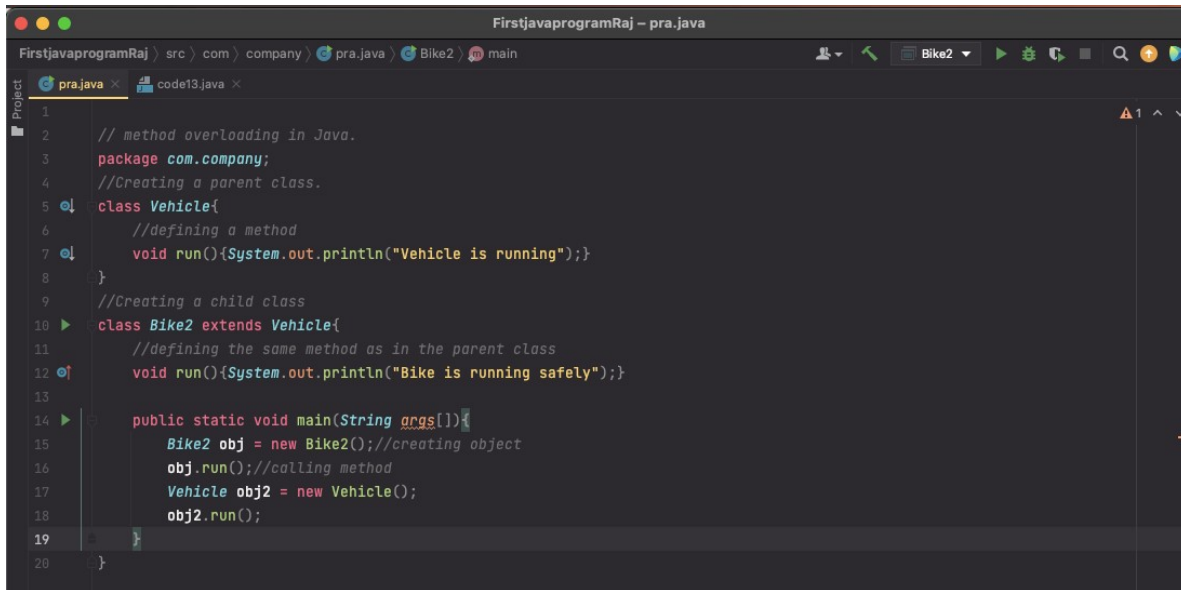
If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in Java**.

Code in text –

```
// method overloading in Java.
package com.company;
//Creating a parent class.
class Vehicle{
    //defining a method
    void run(){System.out.println("Vehicle is running");}
}
//Creating a child class
class Bike2 extends Vehicle{
    //defining the same method as in the parent class
    void run(){System.out.println("Bike is running safely");}

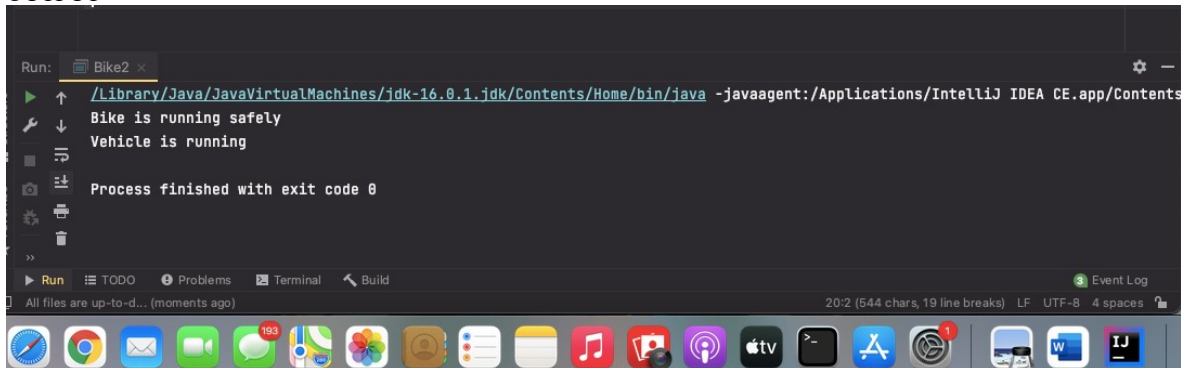
    public static void main(String args[]){
        Bike2 obj = new Bike2();//creating object
        obj.run();//calling method
        Vehicle obj2 = new Vehicle();
        obj2.run();
    }
}
```

Code in Compiler -



```
1 // method overloading in Java.
2 package com.company;
3 //Creating a parent class.
4 class Vehicle{
5     //defining a method
6     void run(){System.out.println("Vehicle is running");}
7 }
8 //Creating a child class
9 class Bike2 extends Vehicle{
10    //defining the same method as in the parent class
11    void run(){System.out.println("Bike is running safely");}
12 }
13
14 public static void main(String args[]){
15     Bike2 obj = new Bike2();//creating object
16     obj.run();//calling method
17     Vehicle obj2 = new Vehicle();
18     obj2.run();
19 }
20 }
```

OUTPUT -



```
Run: Bike2 x
/Library/Java/JavaVirtualMachines/jdk-16.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents
Bike is running safely
Vehicle is running
Process finished with exit code 0
```

Learning outcomes (What I have learnt):

- 1.
- 2.
- 3.
- 4.
- 5.

I

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			