

3. OVERLOADED  
CONSTRUCTOR

Experiment no. 3

date: 23/7/24

roll no: 8014

Aim: To define a class having overloaded constructors and instantiating objects of the same class.

Theory:

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling a constructor, memory for the object is allocated in the memory.

A constructor in Java is a special method that is used to initialise objects.

Syntax: public class MyClass

```
    {  
        MyClass()  
    }
```

Constructors overloading in Java:

- In Java, we can overload constructors like methods.
- The constructor overloading is the concept of having more than one constructors with different parameters so that every constructor can perform a different task.

Syntax: public class name

```
    {  
        name () {  
        }
```

```
        name (String name)
```

```
    }
```

name (String name, int serial no)

}

### Types of constructors:

1. Default Constructors: A constructor that has no parameters is known as default constructor.  
eg:- `new Car();`

2. Parameterised Constructors: A constructor that has parameters is known as parameterised constructor.  
eg:- `new Car (String brand, String model, int year);`

### Procedure:

1. WTP to create a class with 3 constructors overloaded in it. Also show how objects are created using each constructor.

class car {

private String brand;

private String model;

private int year;

public car (String brand, String model, int year) {

this.brand = brand;

this.model = model;

this.year = year;

}

public car (String brand, String model) {

this.brand = brand;

this.model = model;

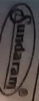
this.year = -1;

}

```
public car () {  
    this.brand = "Unknown";  
    this.model = "Unknown";  
    this.year = -1;  
}
```

```
}  
public void displayingInfo () {  
    System.out.println("car brand: " + brand);  
    System.out.println("car model: " + model);  
    if (year != -1)  
        System.out.println("car year" + year);  
    else  
        System.out.println("car year unknown");  
}
```

```
public class constructor {  
    public static void main (String [] args) {  
        car car1 = new car ("Toyota", "Camry", 2022);  
        car car2 = new car ("Honda", "Civic");  
        car car3 = new car ();  
        System.out.println ("car 1 information");  
        car1.displayingInfo ();  
        System.out.println ("car 2 information");  
        car2.displayingInfo ();  
        car3.displayingInfo ();  
        System.out.println ("car 3 information");  
    }  
}
```



2. Program 2:

```
import java.util.Scanner;
class Student {
```

```
String name;
String dept;
int rollno;
```

```
public Student (String n, String d, int r) {
```

```
this.name = n;
this.dept = d;
this.rollno = r;
```

```
}
public void display ()
```

```
{
    System.out.println ("Name: " + name);
    System.out.println ("Department: " + dept);
    System.out.println ("Roll no: " + rollno);
}
```

```
}
public class {
```

```
public static void main (String [] args) {
```

```
Scanner sc = new Scanner (System.in);
```

```
System.out.println ("enter name:");
```

```
String name = sc.nextLine();
```

```
System.out.println ("enter department:");
```

```
String dept = sc.nextLine();
```

```
System.out.println ("enter roll no:");
```

```
int roll = sc.nextInt();
```

```
Student s1 = new Student (name, dept, roll);
```

```
s1.display ();
}
```

output:-

java : constructor "newCar()" : brand : null

car 1 information : "newCar()" : brand : null

car brand : Toyota

car model : Camry

car year : 2022

car 2 information : "newCar()" : brand : null

car brand : Honda

car model : Civic

car year : -1

car 3 information

car brand : unknown

car model : unknown

car year : unknown

car 1 information : "newCar()" : brand : null

car brand : Toyota

car model : Camry

car year : 2022

car 2 information : "newCar()" : brand : null

car brand : Honda

car model : Civic

car year : -1

car 3 information

car brand : unknown

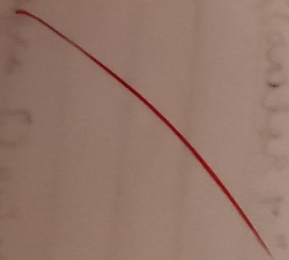
car model : unknown

car year : unknown

Output:-

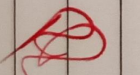
```
javac stud
Enter name:
sudhara
Enter department:
CSE
Enter roll no:
14

Name: sudhara
department: CSE
roll no: 14
```



Conclusion: We have implemented constructor overloading in Java and we have also performed instantiation of objects in the same class.

(7)

  
30/7/08