

Practical No. 06

Aim: Create, debug and run java programs based on classes with objects, method overloading and constructor overloading.

Practical No. 06

Aim: Create, debug and run java programs based on classes with objects, method overloading and constructor overloading.

Theory:

What is a class in Java?

- i) Classes are the building blocks of programs built using the object oriented methodology.
- ii) Objects have certain similar traits - state and behaviour.
- iii) The commonality is provided in a blueprint or template for the instantiation of all similar objects.
- iv) This blueprint is known as a class.

• Every class in Java can be composed of the following elements:

- i) Fields, member variables or instance variables
- ii) member methods or instance methods.
- iii) static or class fields
- iv) static or class methods
- v) Inner class.

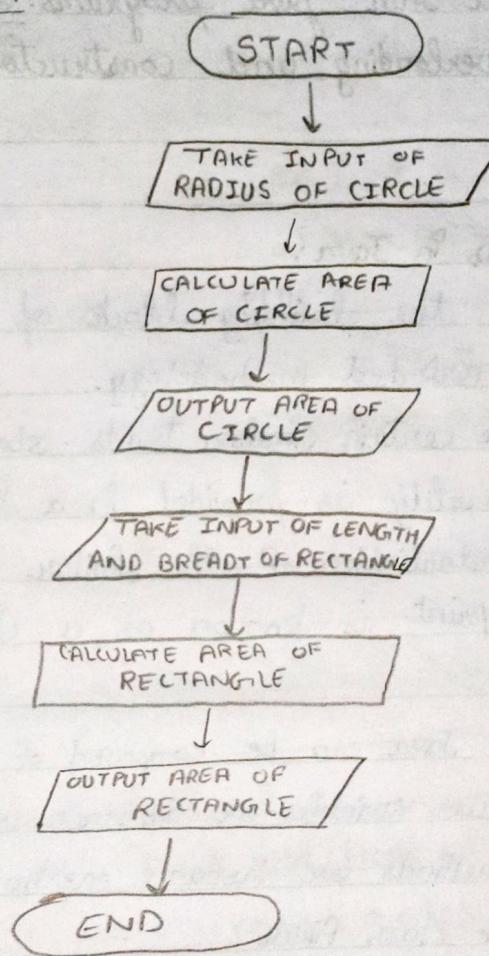
Method overloading:

If a class has multiple methods having same name but different in parameters, it is known as method overloading.

There are two ways to overload the method in Java:

- i) By changing number of arguments
- ii) By changing the data type

Flow chart:



Code 1: Classes, objects, Method Overloading

i) By changing number of arguments:

```
class Addition {  
    int add (int a, int b) {  
        return a+b;  
    }
```

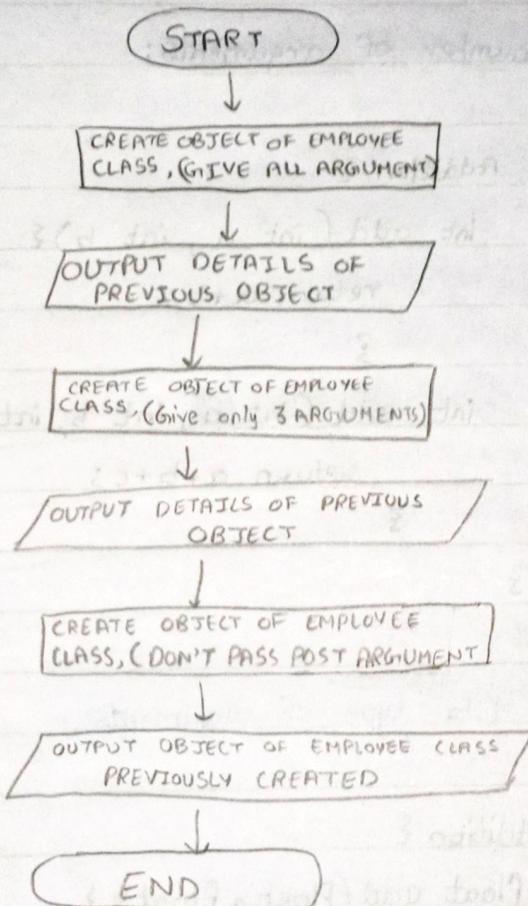
```
    int add (int a, int b, int c) {  
        return a+b+c;  
    }
```

ii) By changing data type of arguments.

```
class addition {  
    float add (float a, float b) {  
        return a+b;  
    }
```

```
    int add (int a, int b) {  
        return a+b;  
    }
```

```
}
```



Code 2 : Constructor Overloading

Constructor overloading:

Constructor overloading in Java is a technique of having more than one constructor with different parameter lists.

These constructors are differentiated by the compiler by the number of parameters in the list and their types.

Example:

```
class STAFF {
```

```
    String Id;
```

```
    String name;
```

```
    int age;
```

```
    STAFF() {
```

```
        age = 0;
```

```
        Id = "00000000";
```

```
        name = "User-none";
```

```
}
```

```
    staff(String name, String Id, int age) {
```

```
        this.name = name;
```

```
        this.id = Id;
```

```
        this.age = age;
```

```
}
```

Suppose we want to implement a function which will return a string. In this situation, there will be two possibilities. One possibility is to overload the function so that it can accept both integer and string type arguments.

3. Time class

: b1.print()

: main.print()

: print(b)

3. Grade

: D - 20%

: "00000000" - b1

: "non-vg" - main

Conclusion:

Hence, by performing this practical I learnt about the various concepts of classes, objects, method overloading and constructor overloading. I also created, debugged and executed java programs based on the above concepts.

Code:

```
import java.util.Scanner;

class calculations{

    final float PI = 3.142f ;

    float area(float radius){
        return radius*radius*PI;
    }

    float area(float length,float breadth){
        return length * breadth;
    }

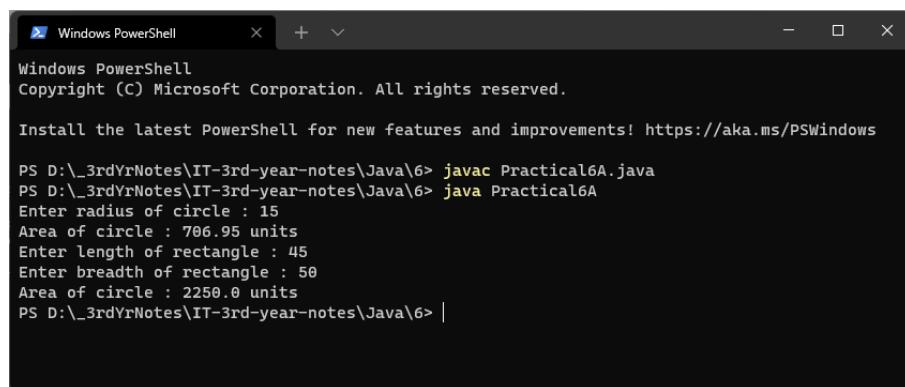
}

class Practical6A{

    public static void main(String[] args){
        calculations obj =new calculations();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter radius of circle : ");
        float radius = sc.nextFloat();
        System.out.println("Area of circle : " + obj.area(radius) + " units"
);
        System.out.print("Enter length of rectangle : ");
        float length = sc.nextFloat();
        System.out.print("Enter breadth of rectangle : ");
        float breadth = sc.nextFloat();
        System.out.println("Area of circle : " + obj.area(length,breadth) + " units" );
    }

}
```

Output:



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the command "javac Practical6A.java" being run, followed by the output of the Java application. The application prompts for the radius of a circle and the length and breadth of a rectangle, then prints their respective areas.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\_3rdYrNotes\IT-3rd-year-notes\Java\6> javac Practical6A.java
PS D:\_3rdYrNotes\IT-3rd-year-notes\Java\6> java Practical6A
Enter radius of circle : 15
Area of circle : 706.95 units
Enter length of rectangle : 45
Enter breadth of rectangle : 50
Area of circle : 2250.0 units
PS D:\_3rdYrNotes\IT-3rd-year-notes\Java\6> |
```

Code:

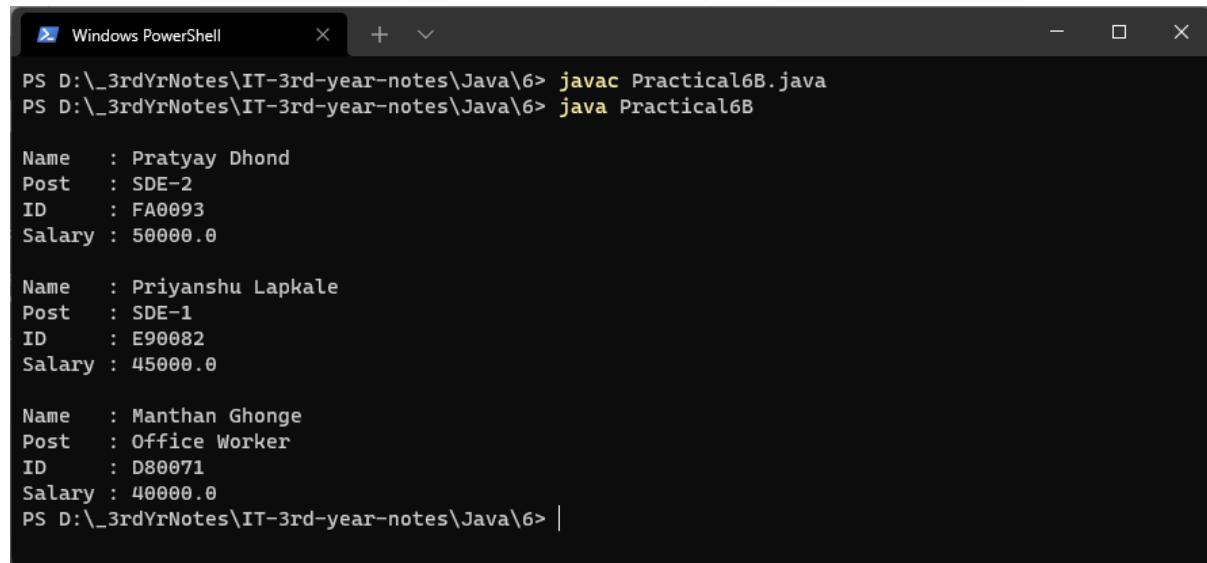
```
class Employee{  
    String name;  
    float Salary;  
    String position;  
    String id;  
  
    Employee(String name, float Salary, String position, String id){  
        this.name = name;  
        this.Salary = Salary;  
        this.position = position;  
        this.id = id;  
    }  
  
    Employee(String name, float Salary, String id){  
        this.name = name;  
        this.Salary = Salary;  
        this.position = "Office Worker";  
        this.id = id;  
    }  
  
    Employee(String name, String position, String id){  
        this.name = name;  
        this.Salary = 45000;  
        this.position = position;  
        this.id = id;  
    }  
  
    void display(){  
        System.out.println();  
        System.out.println("Name : " + name);  
        System.out.println("Post : " + position);  
        System.out.println("ID : " + id);  
        System.out.println("Salary : " + Salary);  
    }  
}  
  
class Practical6B{  
  
    public static void main(String[] args){  
        Employee e1 = new Employee("Pratyay Dhond",50000,"SDE-2","FA0093");  
        e1.display();  
  
        Employee e2 = new Employee("Priyanshu Lapkale","SDE-1","E90082");  
        e2.display();  
    }  
}
```

```
Employee e3 = new Employee("Manthan Ghonge",40000,"D80071");
e3.display();

}

}
```

Output:



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the command "javac Practical6B.java" followed by the output of the "java Practical6B" command. The output displays three employee records with their details: Pratyay Dhond, Priyanshu Lapkale, and Manthan Ghonge.

```
PS D:\_3rdYrNotes\IT-3rd-year-notes\Java\6> javac Practical6B.java
PS D:\_3rdYrNotes\IT-3rd-year-notes\Java\6> java Practical6B

Name    : Pratyay Dhond
Post    : SDE-2
ID      : FA0093
Salary  : 50000.0

Name    : Priyanshu Lapkale
Post    : SDE-1
ID      : E90082
Salary  : 45000.0

Name    : Manthan Ghonge
Post    : Office Worker
ID      : D80071
Salary  : 40000.0
PS D:\_3rdYrNotes\IT-3rd-year-notes\Java\6> |
```

Conclusion:

Hence, by performing this practical I learnt about the various concepts of classes, objects, method overloading and constructor overloading. I also created, debugged and executed java programs based on the above concepts.