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.

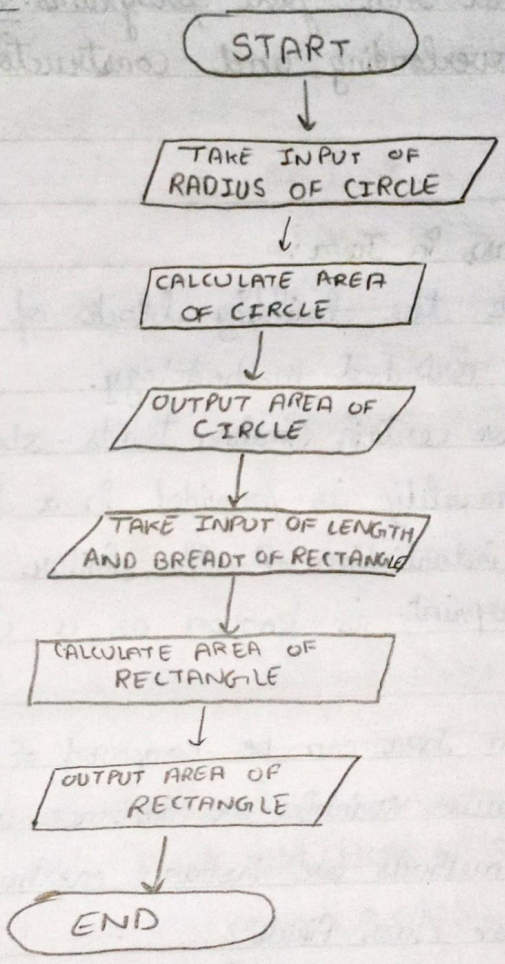
- Every class in Java can be composed of the following elements:
  i) Fields, member variables or instance variables
  ii) member methods or instance methods.
  iii) Static or class fields
  iv) Static or class methods
  v) Inner class.

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:
1) By changing number of arguments
2) By changing the data type

Flow charts:



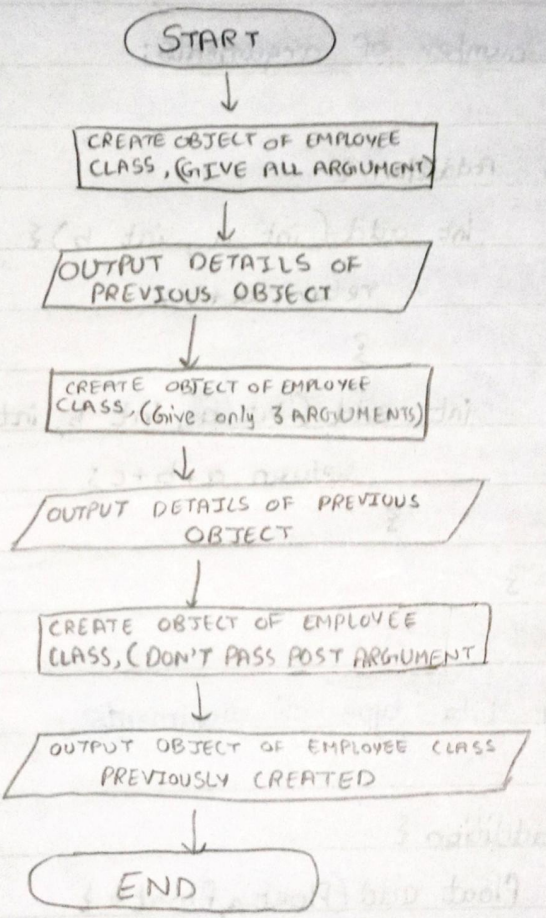
Code 1: Classes, objects, Method Overloading

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, Strid Id, & int age ) {
    this.name = name;
    this.id = Id;
    this.age = age;
}
```

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.