Idris Ratlamwala
C31 - 2003145

Experiment 4: Programs on class and objects

Theory :

Class :
A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. **Modifiers**: A class can be public or has default access.
2. **class keyword:** class keyword is used to create a class.
3. **Class name:** The name should begin with an initial letter (capitalized by convention).
4. **Body:** The class body surrounded by braces, { }.

Constructors are used for initializing new objects. Fields are variables that provides the state of the class and its objects, and methods are used to implement the behavior of the class and its objects.

Object :
It is a basic unit of Object-Oriented Programming and represents the real life entities. A typical Java program creates many objects, which interact by invoking methods. An object consists of :

1. **State**: It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity**: It gives a unique name to an object and enables one object to interact with other objects.

Declaring Objects :
When an object of a class is created, the class is said to be **instantiated**. All the instances share the attributes and the behavior of the class. But the values of those attributes, i.e. the state are unique for each object. A single class may have any number of instances.

Example,
Employee e1;
If we declare reference variable (e1) like this, its value will be undetermined (null) until an object is actually created and assigned to it. Simply declaring a reference variable does not create an object.

Initializing an object :
The new operator instantiates a class by allocating memory for a new object and returning a reference to that memory. The new operator also invokes the class constructor which is generally used to initialize object.

Hence correct way of creating the object is,
Employee e1 = new Employee();

A.

Aim : WAP to read and display details of an employee using single class and its object .

Program :

```
import java.util.Scanner;

public class employee {

    public static void main(String[] args) {
        Emp e1= new Emp();
        e1.register();
        e1.print();
    }
}

class Emp{
    int id,salary;
    String name;
    Scanner sc= new Scanner(System.in);
    void register() {
        System.out.println("Enter name:");
        name = sc.nextLine();
        System.out.println("\nEnter id number:");
        id = sc.nextInt();
        System.out.println("\nEnter salary:");
        salary = sc.nextInt();
    }
    void print(){
        System.out.println("\nName: "+name+"\nId: "+id+"\nSalary: "+salary);
    }
}
```

Output :

```
Enter name:
Idris Ratlamwala

Enter id number:
2003145

Enter salary:
80000

Name: Idris Ratlamwala
Id: 2003145
Salary: 80000
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs\expt4>
```

B.

Aim : WAP to find maximum of three numbers using conditional operator, using two classes and function returning result .

Program :

```java
import java.util.Scanner;

public class maxByclass {

    public static void main(String[] args) {
        Max.get();
        System.out.println("Max : "+ Max.calcMax());
    }
}

class Max{
    static int n1;
    static int n2;
    static int n3;
    static void get() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter 3 numbers : ");
        n1=sc.nextInt();
        n2=sc.nextInt();
        n3=sc.nextInt();
    }
    static int calcMax(){
        return (n1>n2&&n1>n3)?n1:((n2>n3)?n2:n3);
    }
}
```

Output :

```
Enter 3 numbers :
3 9 1
Max : 9
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs\expt4> java maxByclass.java
Enter 3 numbers :
29 11 85
Max : 85
PS C:\Users\IsmailRatlamwala\Documents\College prog\Oops Labs\expt4>
```