**Experiment number 2**

**Aim:** Write a class ‘Employee’ with the following specifications:

Data Members: String empName : Name of the employee

String empId : Unique ID of the employee

salary : Salary of employee (Choose appropriate data-type)

Create a constructor to define the values for these data members. Create another driver class ‘EmployeeDemo’ with a main() method, which creates a new Employee object for an employee named “Raj” with Unique ID “E201945”, salary 12,000. Print these details to the console. Details to be taken by the user.

**Theory:** In this program, we have to use encapsulation. Encapsulation is defined as the wrapping up of data into a single unit.

**Algorithm:**

**Code:**

import java.util.\*;

class Employee{

private String empName,empId;

private float salary;

Employee(){ //constructor

empName="Samarth";

empId="E202023";

salary=15000f;

}

public String getName(){

return empName;

}

public String getId(){

return empId;

}

public float getSalary(){

return salary;

}

public void setName(String newName) {

this.empName = newName;

}

public void setId(String newId){

this.empId = newId;

}

public void setSalary(int newSal){

this.salary = newSal;

}

public void display(String a, String b , float c){

System.out.println(a);

System.out.println(b);

System.out.println(c);

}

}

public class EmployeeDemo{ //driver class

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("enter the number of employees=");

int n= sc.nextInt();

Employee[] arr =new Employee[n];

for(int i = 0; i<n ; i++){

arr[i]=new Employee();

String newName=sc.next();

String newId=sc.next();

int newSal=sc.nextInt();

//Employee emp= new Employee();

arr[i].setName(newName);

arr[i].setId(newId);

arr[i].setSalary(newSal);

String a=arr[i].getName();

String b=arr[i].getId();

float c=arr[i].getSalary();

arr[i].display(a,b,c);

}

}

}

**Output:**

****

**Conclusion:**

by writing this program, I learnt how to use encapsulation. The concept of public and private access specifiers became more clear to me, and i also learnt how get and set methods work in java.