25 Sept 2020 Rinoy Kuriyakose R3 56

**Experiment:6**

**Aim:**

Write two Java classes Employee and Engineer. Engineer should inherit from Employee class. Employee class to have two methods display() and calcSalary(). Write a program to display the engineer salary and to display from Employee class using a single object instantiation (i.e., only one object creation is allowed). ●display() only prints the name of the class and does not return any value. Ex. “ Name of class is Employee.” ● calcSalary() in Employee displays “Salary of employee is 10000” and calcSalary() in Engineer displays “Salary of employee is 20000.”

**Concepts Used:**

Inhertitance, static keyword.

**Algorithms:**

Class Employee

1. Declare fields salary and className

2. Define display method to print className

3. Define static calcSalary method to print salary

Class Engineer inherits Class Employee

1. Call static methods display and calcSalary

2. Create Engineer object and call methods display and calcSalary on the object

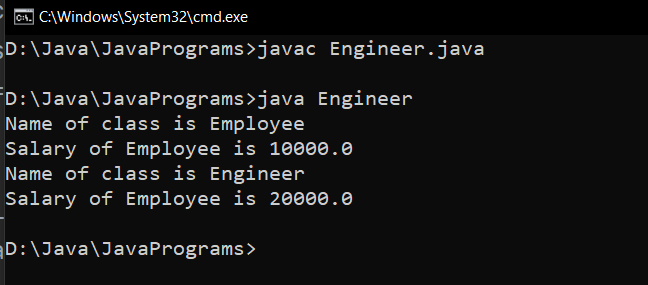
**Program:**

class Employee {

static double salary;  
 static String classname;  
 static void display(String clsname) {  
 classname = clsname;  
 System.out.println("Name of class is " + classname);  
 }  
 static void calcSalary(double sal) {  
 salary = sal;  
 System.out.println("Salary of Employee is " + sal);  
 }  
}  
class Engineer extends Employee {

public static void main(String args[]){  
 Employee.display("Employee");  
 Employee.calcSalary(10000);  
 Engineer obj = new Engineer();  
 obj.display("Engineer");  
 obj.calcSalary(20000);  
 }  
}

**Output:**



**Result:**

Static methods of parent class Employee are accessed with and without creating objects from child class Engineer.

25 Sept 2020 Rinoy Kuriyakose R3 56

**Experiment:7**

**Aim:**

Write a java program to create an abstract class named Shape that contains an empty method named numberOfSides( ). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures.

**Concepts Used:**

Inheritance, Method overriding, Data abstraction using abstract keyword.

**Algorithm:**

Abstract Class Shape

1. Declare abstract method numOfSides()

Class Triangle inherits Class Shape

1. Override numOfSides() method and return 3

Class Rectangle inherits Class Shape

1. Override numOfSides ()method and return 4

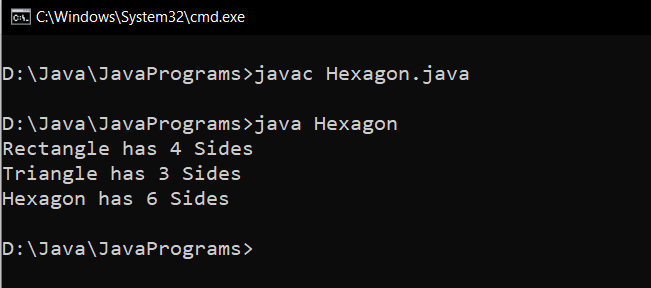
Class Hexagon inherits Class Shape

1. Override numOfSides() method and return 6

**Program:**

abstract class Shape{  
 abstract void numberOfSides();  
}  
class Triangle extends Shape{  
 void numberOfSides(){  
 System.out.println("Triangle has 3 Sides");  
 }  
}  
class Rectangle extends Shape{  
 void numberOfSides(){  
 System.out.println("Rectangle has 4 Sides");  
 }  
}  
class Hexagon extends Shape{  
 void numberOfSides(){  
 System.out.println("Hexagon has 6 Sides");  
 }  
 public static void main(String[] args){  
 Rectangle rect = new Rectangle();  
 Triangle tri = new Triangle();  
 Hexagon hex = new Hexagon();  
 rect.numberOfSides();  
 tri.numberOfSides();  
 hex.numberOfSides();  
 }  
}

**Output:**

****

**Result:**

Data abstraction and method overriding are performed on different shapes.