Lab – 8

1. Write an abstract class that contains basic details of employee namely name and empid and a concrete method commonEmpDetaills() to display the basic Details of an employee. Include another abstract method confidentialDetails() to display confidential details. Extend the abstract class in another class HR with employee confidential details like Salary and Performance and display them in the implementation of the abstract method confidentialDetails() .

Program :

```
public abstract class EmployeeDetails {
    String name;
    int empid;
    EmployeeDetails(String name,int empid){
            this.name=name;
            this.empid=empid;
    }
    public abstract void confidentialDetails();
    public void commonEmpDetails() {
            System.out.println("Name :"+name);
            System.out.println("Emp Id: "+empid);
    }
}
public class HR extends EmployeeDetails {
            double salary;
            String performance;
            HR(String name,int empid,double salary,String performance){
                    super(name,empid);
                    this.salary=salary;
                    this.performance=performance;
            }
            public void confidentialDetails() {
                    System.out.println("Salary :"+salary);
```

```
                System.out.println("Performance :"+performance);
        }
    }
    public class Demo {
        public static void main(String []args) {
            HR h1=new HR("Aravind",79992,200000,"Excellent");
            h1.commonEmpDetails();
            h1.confidentialDetails();
        }
    }
```

2. Define a class Person with name, age as attributes and an abstract method occupation () that returns a string. The classes Student, Employee and SeniorCitizen inherit Person with occupations as studying, working and retired, respectively. The main () method must create the objects based on age. (If age is <=25, then Student, if age is between 26 to 60 then Employee, else senior). The data must be stored in an array of Person (assume size of 10) and must display all data.

Program :

```
public abstract class Person {
    protected String name;
    protected int age;
    public Person(String name,int age) {
        this.name=name;
        this.age=age;
    }
    public abstract String occupation();
}
public class Student extends Person {
    public Student(String name,int age) {
        super(name,age);
    }
```

```
	public String occupation() {
		return "Studying";
	}
}
public class Employee extends Person{
		public Employee(String name,int age) {
			super(name,age);
		}
		public String occupation() {
			return "Working";
		}
}
public class SeniorCitizen extends Person {
	public SeniorCitizen(String name,int age) {
		super(name,age);
	}
	public String occupation() {
		return "Retired";
	}
}
import java.util.Scanner;
public class Demo {
		public static void main(String []args) {
			Scanner sc=new Scanner(System.in);
			Person p[]=new Person[10];
			int age=0;
			String name="";
			for(int i=0;i<3;i++) {
				System.out.println("Enter name and age");
				name=sc.next();
				age=sc.nextInt();
				if(age<=25) {
					p[i]=new Student(name,age);
				}
```

```
                else if(age<=60) {
                        p[i]= new Employee(name,age);
                }
                else {
                        p[i]= new SeniorCitizen(name,age);
                }
        }
        for(int i=0;i<10;i++) {
                System.out.println(p[i].occupation());
        }
        sc.close();
}}
```