

B.hanumanthu ISTE60

1)

```
interface Drivable{
    public void start();
    public void stop();
}

interface Flyable{
    public void takeOff();
    public void land();
}

class Car implements Drivable{
    public void start(){
        System.out.println("Car Started..");
    }
    public void stop(){
        System.out.println("Car Stopped..");
    }
}

class AirCraft implements Flyable{
    public void takeOff(){
        System.out.println("AirCraft takeoff...");
    }
    public void land(){
        System.out.println("AirCraft Landed");
    }
}

public class Vehicle{
    public static void main(String args[]){
        //Creating and Calling Objects
        System.out.println("Calling Car");
        Car c = new Car();
    }
}
```

```

        c.start();

        c.stop();

        System.out.println("Calling AirCraft");

        AirCraft a = new AirCraft();

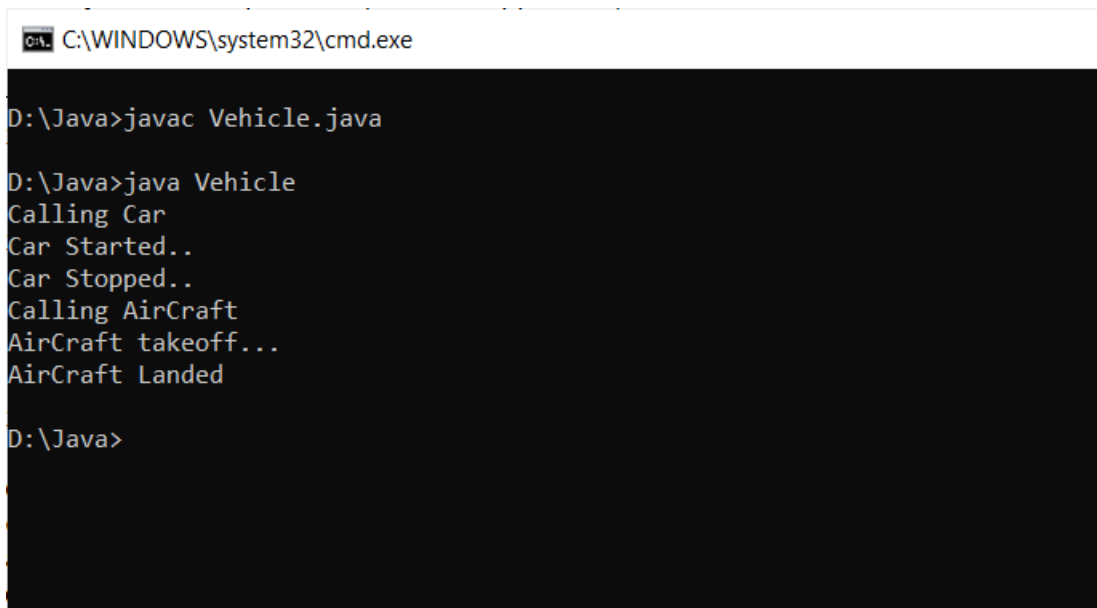
        a.takeOff();

        a.land();

    }

}

```



```

C:\WINDOWS\system32\cmd.exe

D:\Java>javac Vehicle.java

D:\Java>java Vehicle
Calling Car
Car Started..
Car Stopped..
Calling AirCraft
AirCraft takeoff...
AirCraft Landed

D:\Java>

```

2)

```

interface AudioPlayer{

    public void playAudio();

}

interface VideoPlayer{

    public void playVideo();

}

public class Media implements AudioPlayer, VideoPlayer{

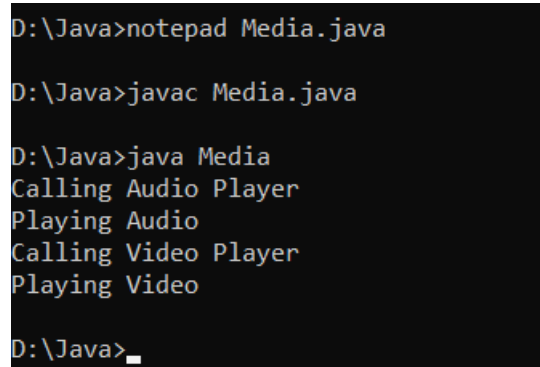
    public void playAudio(){

```

```

        System.out.println("Playing Audio");
    }
    public void playVideo(){
        System.out.println("Playing Video");
    }
    static public void main(String args[]){
        Media m = new Media();
        System.out.println("Calling Audio Player");
        m.playAudio();
        System.out.println("Calling Video Player");
        m.playVideo();
    }
}

```



```

D:\Java>notepad Media.java

D:\Java>javac Media.java

D:\Java>java Media
Calling Audio Player
Playing Audio
Calling Video Player
Playing Video

D:\Java>_

```

4.

```

interface EditorPlugin{
    public void performAction();
    public void DataEntry();
    public void DBConnect();
}

public class Plugin implements EditorPlugin{
    public void performAction(){
        System.out.println("Performing Action..");
    }
}

```

```

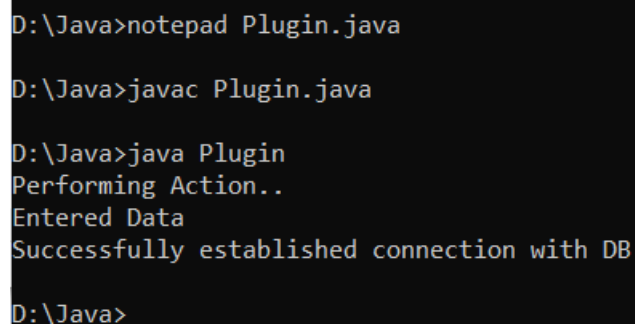
    }

    public void DataEntry(){
        System.out.println("Entered Data");
    }

    public void DBConnect(){
        System.out.println("Successfully established connection with DB");
    }

    static public void main(String args[]){
        Plugin p = new Plugin();
        p.performAction();
        p.DataEntry();
        p.DBConnect();
    }
}

```



```

D:\Java>notepad Plugin.java

D:\Java>javac Plugin.java

D:\Java>java Plugin
Performing Action..
Entered Data
Successfully established connection with DB

D:\Java>

```

5.

```

interface RemoteControl{
    public void powerOn();
    public void powerOff();
}

class TVRemote implements RemoteControl{
    public void powerOn(){
        System.out.println("TV switched ON");
    }
}

```

```

        public void powerOff(){
            System.out.println("TV switched OFF");
        }
    }

    class ACRemote implements RemoteControl{
        public void powerOn(){
            System.out.println("AC switched ON");
        }
        public void powerOff(){
            System.out.println("AC switched OFF");
        }
    }

    public class Control{
        static public void main(String args[]){
            System.out.println("Calling TV");
            TVRemote t = new TVRemote();
            t.powerOn();
            t.powerOff();
            System.out.println("Calling AC");
            ACRemote a = new ACRemote();
            a.powerOn();
            a.powerOff();
        }
    }
}

```

```
D:\Java>notepad Control.java
```

```
D:\Java>javac Control.java
```

```
D:\Java>java Control
```

```
Calling TV
```

```
TV switched ON
```

```
TV switched OFF
```

```
Calling AC
```

```
AC switched ON
```

```
AC switched OFF
```

```
D:\Java>
```

6.

```
class Animal{
```

```
    void eat(){
```

```
        System.out.println("It can Eat");
```

```
    }
```

```
    void walk(){
```

```
        System.out.println("It can Walk");
```

```
    }
```

```
    void Fly(){
```

```
        System.out.println("It can Fly");
```

```
    }
```

```
}
```

```
class Mammal extends Animal{
```

```
    void eat(){
```

```
        System.out.println("Mammal can Eat");
```

```
    }
```

```
    void walk(){
```

```
        System.out.println("Mammal can Walk");
```

```
    }
```

```
}
```

```
class Bird extends Animal{
```

```

        void eat(){
            System.out.println("Bird can Eat");
        }
        void Fly(){
            System.out.println("Bird can Fly");
        }
    }

    class Reptile extends Animal{
        void eat(){
            System.out.println("Reptile can eat");
        }
        void walk(){
            System.out.println("Reptile can walk");
        }
    }

    class Test{
        public static void main(String args[]){
            Animal a;
            a = new Bird();
            a.eat();
            a.Fly();
            a = new Mammal();
            a.eat();
            a.walk();
            a = new Reptile();
            a.eat();
            a.walk();
        }
    }
}

```

```
D:\Java>javac Test.java
```

```
D:\Java>java Test
```

```
Bird can Eat
```

```
Bird can Fly
```

```
Mammal can Eat
```

```
Mammal can Walk
```

```
Reptile can eat
```

```
Reptile can walk
```

```
D:\Java>
```

7.

```
class Shape{
```

```
    void draw(){
```

```
        System.out.println("Successfully Drawn");
```

```
    }
```

```
}
```

```
class Circle extends Shape{
```

```
    void draw(){
```

```
        System.out.println("Successfully Drawn Circle");
```

```
    }
```

```
}
```

```
class Rectangle extends Shape{
```

```
    void draw(){
```

```
        System.out.println("Successfully Drawn Rectangle");
```

```
    }
```

```
}
```

```
class Triangle extends Shape{
```

```
    void draw(){
```

```
        System.out.println("Successfully Drawn Triangle");
```

```
    }
```

```
}
```

```
class TestShape{
```



```

    public static void main(String args[]){

        Shape s;

        s = new Circle();

        s.draw();

        s = new Rectangle();

        s.draw();

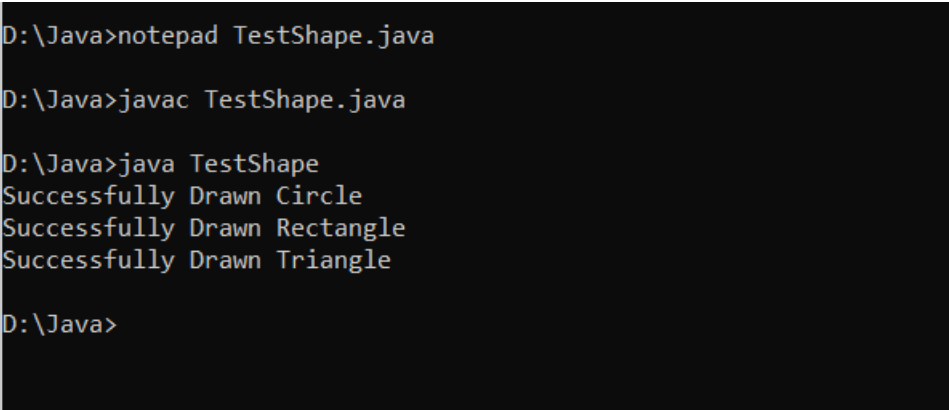
        s = new Triangle();

        s.draw();

    }

}

```



```

D:\Java>notepad TestShape.java

D:\Java>javac TestShape.java

D:\Java>java TestShape
Successfully Drawn Circle
Successfully Drawn Rectangle
Successfully Drawn Triangle

D:\Java>

```

8.

```

class vehicle{

    void speed(int limit){

        System.out.println("Speed Limit: "+limit);

    }

}

class Car extends vehicle{

    void speed(int limit){

        System.out.println("Car Speed Limit: "+limit);

    }

}

```

```

}

class Motorcycle extends vehicle{
    void speed(int limit){
        System.out.println("Motorcycle Speed Limit: "+limit);
    }
}

class Truck extends vehicle{
    void speed(int limit){
        System.out.println("Truck Speed Limit: "+limit);
    }
}

class TestVehicle{
    public static void main(String args[]){
        vehicle v;
        v = new Car();
        v.speed(80);
        v = new Motorcycle();
        v.speed(60);
        v = new Truck();
        v.speed(70);
    }
}

```

```

D:\Java>notepad TestVehicle.java

D:\Java>javac TestVehicle.java

D:\Java>java TestVehicle
Car Speed Limit: 80
Motorcycle Speed Limit: 60
Truck Speed Limit: 70

D:\Java>_

```

9.

```
class Emp{
    void workingHour(int time){
        System.out.println("Total Working Hours: "+time);
    }
}

class Manager extends Emp{
    void workingHour(int time){
        System.out.println("Manager Total Working Hours: "+time);
    }
}

class Developer extends Emp{
    void workingHour(int time){
        System.out.println("DeveloperTotal Working Hours: "+time);
    }
}

class SalesPerson extends Emp{
    void workingHour(int time){
        System.out.println("Sales Person Total Working Hours: "+time);
    }
}

class Employee{
    public static void main(String args[]){
        Emp e;
        e = new Manager();
        e.workingHour(6);

        e = new Developer();
        e.workingHour(8);
    }
}
```

```
        e = new SalesPerson();  
        e.workingHour(10);  
    }  
}
```

```
D:\Java>javac Employee.java  
  
D:\Java>java Employee  
Manager Total Working Hours: 6  
DeveloperTotal Working Hours: 8  
Sales Person Total Working Hours: 10  
  
D:\Java>
```