

1) package transport

```
class Vehicle {
    String model;
    long regNo;
    double vehSpeed;
    double fuelCap;
    double fConsumption;

    Vehicle() {
    }

    Vehicle(String model, long regNo, double vehSpeed, double fuelCap,
            double fConsumption) {
        this.model = model;
        this.regNo = regNo;
        this.vehSpeed = vehSpeed;
        this.fuelCap = fuelCap;
        this.fConsumption = fConsumption;
    }

    public void setModel(String model) { this.model = model; }
    public void setRegNo(long regNo) { this.regNo = regNo; }
    public void setVehSpeed(double vehSpeed) { this.vehSpeed = vehSpeed; }
    public void setfConsumption(double fConsumption) { this.fConsumption = fConsumption; }
    public void fuelNeeded(double distance) {
        double fuelInNeed = distance / fConsumption;
        System.out.println("Fuel Needed is: " + fuelInNeed);
    }
    public void distanceCovered(double time) {
        double distCovered = time * vehSpeed;
        System.out.println("The distance covered is: " + distCovered);
    }
    void display() {
        System.out.println("Vehicle model: " + model);
        S.O.P("Registration Number: " + regNo);
        S.O.P("Fuel capacity: " + fuelCap);
    }
}
```



```

    }
    SOP("Fuel Consumption: " + fConsumption);
}

```

```

package Transport;

```

```

public class Truck extends Vehicle {
    double weightLimit;
    Truck() {
    }
    Truck(double weightLimit, String model, long regNo, double speed,
           double fuelCap, double fConsumption) {
    }
    Super(model, regNo, speed, fuelCap, fConsumption);
    this.weightLimit = weightLimit;
    public void setWeightLimit(double weightLimit) {
        this.weightLimit = weightLimit;
    }
    @Override
    void display() { Super.display(); }
}

```

```

package Transport;
public class Bus extends Vehicle {
    double passengers;
    Bus(double passengers, String model, long regNo, double vehSpeed,
        double fuelCap, double fConsumption) {
        super(model, regNo, vehSpeed, fuelCap, fConsumption);
        this.passengers = passengers;
        this.model = model;
        this.vehSpeed = vehSpeed;
        this.fuelCap = fuelCap;
        this.fConsumption = fConsumption;
    }
    public void setPassenger(double passengers) { this.passengers = passengers; }
    @Override
    void display() { super.display();
        SOP("No of passengers: " + passengers);
    }
}

```


package Transport;

```
public class Transport {  
    n.s.m.v (String[] args) {  
        Truck t = new Truck();  
        t.set Model ("Ashoka Leyland");  
        t.set VehSpeed (35);  
        t.set RegNo (1141076);  
        t.set FuelCap (68.42);  
        t.set Consumption (27.0);  
        t.set WeightLimit (2000);
```

```
        t.display();  
        t.fuelNeeded (500);  
        t.distanceCovered (5);
```

```
        Bus b = new Bus();  
        b.set Model ("Volvo");  
        b.set VehSpeed (60);  
        b.set RegNo (1141077);  
        b.set FuelCap (54.25);  
        b.set Consumption (82);  
        b.set Passengers (60);
```

```
        b.display();  
        b.fuelNeeded (275);  
        b.distanceCovered (10);
```

```
    }  
}
```



```

2) import java.lang.Thread;
import java.util.Scanner;

public class Trial {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();

        Printer p = new Printer();

        odd o = new odd(n, p);
        o.setName("Odd: ");

        even e = new even(n, p);
        e.setName("Even: ");

        o.start();
        e.start();
    }
}

class odd extends Thread {
    int limit;
    Printer p;

    public odd(int limit, Scanner Printer p) {
        this.limit = limit;
        this.p = p;
    }

    @Override
    public void run() {
        int oddNum = 1;
        while (oddNum <= limit) {
            p.print(oddNum);
            oddNum += 2;
        }
    }
}

```



```
class even extends Thread {
```

```
    int limit;
```

```
    Printer p;
```

```
    public even(int limit, Printer p) {
```

```
        this.limit = limit;
```

```
        this.p = p; }
```

```
    @Override
```

```
    public void run() {
```

```
        int evenNum = 0;
```

```
        while (evenNum <= limit) {
```

```
            p.printEven(evenNum);
```

```
            evenNum += 2; }
```

```
    }
```

```
}
```

```
class Printer {
```

```
    boolean isOdd = false;
```

```
    synchronized void printOdd(int number) {
```

```
        while (isOdd) {
```

```
            try { wait(); }
```

```
            catch (InterruptedException e) { e.printStackTrace(); }
```

```
        }
```

```
        SOP(Thread.currentThread().getName() + ": " + number);
```

```
        isOdd = true;
```

```
try { try
```

```
        notify();
```

```
    }
```

```
    synchronized void printEven(int number) {
```

```
        while (!isOdd) {
```

```
            try { wait(); }
```

```
            catch (InterruptedException e) { try e.printStackTrace(); }
```

```
        }
```



```

        GOP (Thread - current Thread().getName() + ":" + number);
        isOddPrinted = false;
        notify();
    }
}

```

```

3) import java.awt.*;
import java.applet.*;
import java.util.*;

```

```

public class ClockApplet extends Applet implements Runnable {
    Thread t, t1;
    public void start() {
        t = new Thread(this);
        t.start();
    }
    public void run() {
        t1 = Thread.currentThread();
        while (t1 == t) {
            repaint();
        }
    }
    public void paint(Graphics g) {
        Calendar cal = new GregorianCalendar();
        String hour = String.valueOf(cal.get(Calendar.HOUR));
        String minute = String.valueOf(cal.get(Calendar.MINUTE));
        String second = String.valueOf(cal.get(Calendar.SECOND));
        g.drawString(hour + ":" + minute + ":" + second, 20, 30);
    }
}

```