

```
1  *** SRP ***
2  public class Student{
3      public void printDetails(){
4      public void calculatePercent(){
5      public void addStudent(){
6
7  }
8
9  public class Student{
10     public void addStudent(){
11 }
12 public class PrintStudentDetails{
13     public void printDetails(){
14 }
15
16 public class StudentMetrics{
17     public void calculatePercent(){
18 }
19
20 *** OCP ***
21 public class VehicleInfo{
22     public long vehicleNumber(Vehicle vcl){
23         if(vcl instanceof Car){
24             return vcl.getNumber();
25         }
26         if(vcl instanceof Bike){
27             vcl.getNumber();
28         }
29         if(vcl instanceof Truck){
30
31         }
32 }
33
34 public class VehicleInfo{
35     public long vehicleNumber(){
36         //some functionality
37     }
38 }
39 public class Car extends VehicleInfo{
40     public long vehicleNumber(){
41         return this.vehicleNumber;
```

```
42     }
43 }
44
45 public class Bike extends VehicleInfo{
46     public long vehicleNumber(){
47         return this.vehicleNumber;
48     }
49 }
50 public class Truck extends VehicleInfo{
51     public long vehicleNumber(){
52         return this.vehicleNumber;
53     }
54 }
55
56 *** LSP ***
57 public class Rectangle{
58     private double width;
59     private double height;
60     public double area(){
61     public void setWidth(double width){this.width=width;}
62     public void setHeight(double height){this.height=height;}
63 }
64 public class Square extends Rectangle{
65
66 }
67
68 public class Vehicle{
69     public void startEngine(){
70         //the default implementaion for engine start
71     }
72     public void accelerate(){
73         //default implementation for acceleration
74     }
75 }
76 public class Car extends Vehicle{
77     public void startEngine(){
78         engageIgnition();
79         super.startEngine();
80     }
81     private void engageIgnition(){
82         //ignition procedure
```

```
83     }
84 }
85 public class ElectricCar extends Vehicle{
86     public void accelerate(){
87         increaseVoltage();
88         connectIndiviuallEngines();
89     }
90     private void increaseVoltage(){
91     private void connectIndiviuallEngine(){
92 }
93
94 public class CarDriver{
95     public void drive(Vehicle v){
96         v.startEngine();
97         v.accelerate();
98     }
99 }
100 *** ISP ***
101 public interface Conversion{
102     public void intToDouble();
103     public void intToChar();
104     public void charToString();
105 }
106
107 public interface ConvertIntToDouble{
108     public void intToDouble();
109 }
110 public interface ConvertIntToChar{
111     public void intToChar();
112 }
113 public interface ConvertCharToString{
114     public void charToString();
115 }
116
117 public class DataConversion implements
    ConvertIntToDouble, ConvertCharToString{
118
119 }
120
121 *** DIP ***
122
```

```
123 public class WindowsMachine{}
124
125 public class WindowsMachine{
126     private final StandardKeyboard standardKeyboard;
127     private Monitor monitor;
128     public WindowsMachine(){
129         monitor=new Monitor();
130         standardKeyboard=new StandardKeyboard();
131     }
132 }
133
134
135 public interface Keyboard{}
136 public interface Monitor{}
137
138 public class StandardKeyboard implements Keyboard{
139
140 }
141 public class LedMonitor implements Monitor{
142
143 }
144 public class WindowsMachine{
145     private final Keyboard keyboard;
146     private final Monitor monitor;
147     public WindowsMachine(Keyboard keyboard,Monitor monitor){
148         this.keyboard=keyboard;
149         this.monitor=monitor;
150     }
151 }
```