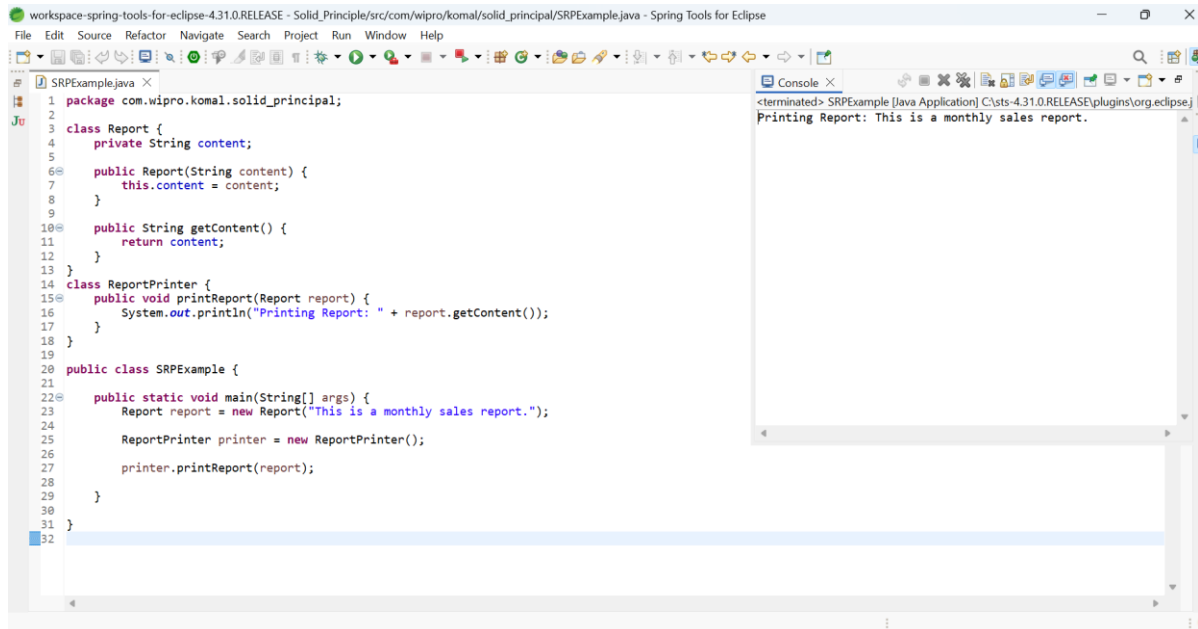


Name:- Komal Sudhakar Baviskar

Solid principal outputs:

1)

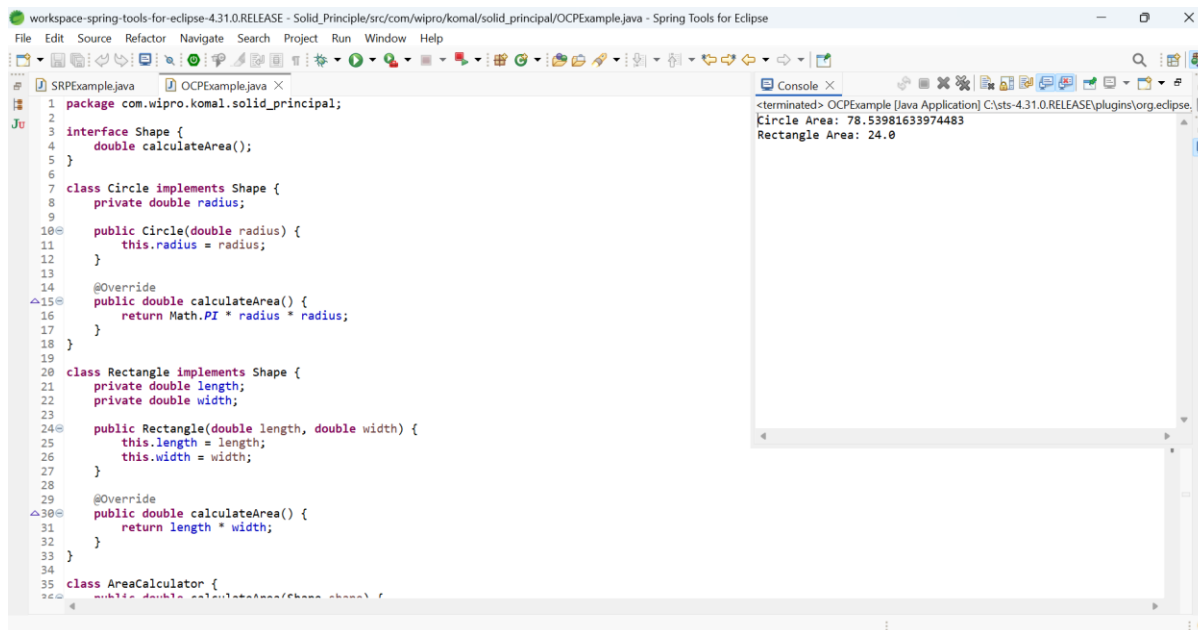


The screenshot shows the Eclipse IDE with a Java project named 'Solid\_Principle'. The file 'SRPExample.java' is open in the editor. The code defines a 'Report' class with a 'content' attribute and a 'ReportPrinter' class that prints the report content. The 'SRPExample' class has a 'main' method that creates a 'Report' object with the content 'This is a monthly sales report.' and a 'ReportPrinter' object, then calls 'printReport' on the printer. The console on the right shows the output: 'Printing Report: This is a monthly sales report.'

```
1 package com.wipro.komal.solid_principal;
2
3 class Report {
4     private String content;
5
6     public Report(String content) {
7         this.content = content;
8     }
9
10    public String getContent() {
11        return content;
12    }
13 }
14 class ReportPrinter {
15     public void printReport(Report report) {
16         System.out.println("Printing Report: " + report.getContent());
17     }
18 }
19
20 public class SRPExample {
21
22     public static void main(String[] args) {
23         Report report = new Report("This is a monthly sales report.");
24
25         ReportPrinter printer = new ReportPrinter();
26
27         printer.printReport(report);
28     }
29 }
30
31
32
```

Console Output:  
<terminated> SRPExample [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.j...  
Printing Report: This is a monthly sales report.

2)



The screenshot shows the Eclipse IDE with a Java project named 'Solid\_Principle'. The file 'OCPExample.java' is open in the editor. The code defines an interface 'Shape' with a 'calculateArea()' method. Two classes, 'Circle' and 'Rectangle', implement the 'Shape' interface. 'Circle' calculates the area using the formula  $\pi r^2$ , and 'Rectangle' calculates the area using the formula  $length \times width$ . The 'AreaCalculator' class has a 'calculateArea' method that takes a 'Shape' object and returns its area. The console on the right shows the output: 'Circle Area: 78.53981633974483' and 'Rectangle Area: 24.0'.

```
1 package com.wipro.komal.solid_principal;
2
3 interface Shape {
4     double calculateArea();
5 }
6
7 class Circle implements Shape {
8     private double radius;
9
10    public Circle(double radius) {
11        this.radius = radius;
12    }
13
14    @Override
15    public double calculateArea() {
16        return Math.PI * radius * radius;
17    }
18 }
19
20 class Rectangle implements Shape {
21     private double length;
22     private double width;
23
24    public Rectangle(double length, double width) {
25        this.length = length;
26        this.width = width;
27    }
28
29    @Override
30    public double calculateArea() {
31        return length * width;
32    }
33 }
34
35 class AreaCalculator {
36     public double calculateArea(Shape shape) {
37
38     }
39 }
40
```

Console Output:  
<terminated> OCPExample [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.j...  
Circle Area: 78.53981633974483  
Rectangle Area: 24.0

3)

The screenshot shows the Eclipse IDE with a Java project named 'Solid\_Principle'. The main editor displays the file 'ISPEExample.java'. The code defines an interface 'Discount' with a method 'apply(double amount)'. Two classes, 'StudentDiscount' and 'RegularDiscount', implement this interface. 'StudentDiscount' applies a 75% discount, and 'RegularDiscount' applies a 50% discount. The 'ISPEExample' class has a 'main' method that creates instances of both discount classes and applies them to a bill of 1205.25. The console output shows the discounted amounts: 903.9375 for StudentDiscount and 602.625 for RegularDiscount.

```
1 package com.wipro.komal.solid_principal;
2
3 interface Discount{
4     void apply(double amount);
5 }
6
7 class StudentDiscount implements Discount{
8     public void apply(double amount) {
9         double Discountamount = amount * 0.75;
10        System.out.println("Discounted Amount is"+ " "+Discountamount);
11    }
12 }
13
14 class RegularDiscount implements Discount{
15     public void apply(double amount) {
16         double Discountamount= amount * 0.5;
17        System.out.println("Discounted Amount is"+ " "+ Discountamount);
18    }
19 }
20
21
22 public class ISPEExample{
23     public static void main(String[] args) {
24         double bill = 1205.25;
25
26         StudentDiscount ds = new StudentDiscount();
27         ds.apply(bill);
28         RegularDiscount dr = new RegularDiscount();
29         dr.apply(bill);
30     }
31 }
```

Console Output:

```
<terminated> ISPEExample [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.j
Discounted Amount is 903.9375
Discounted Amount is 602.625
```

4)

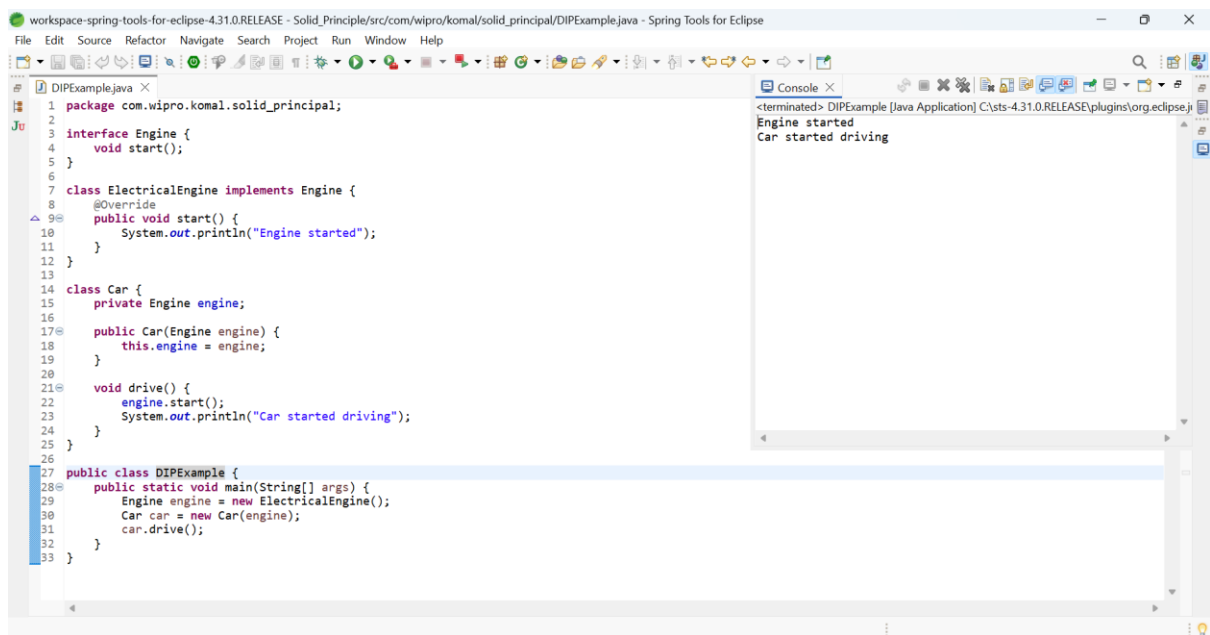
The screenshot shows the Eclipse IDE with the same 'Solid\_Principle' project. The main editor displays 'ISPEExample.java'. The code defines two interfaces: 'Cooking' with a method 'cook()' and 'Cleaning' with a method 'clean()'. Two classes, 'Chef' and 'Cleaner', implement these interfaces. 'Chef' implements 'cook()' by printing 'Chef is cooking', and 'Cleaner' implements 'clean()' by printing 'Cleaner is cleaning'. The 'ISPEExample' class has a 'main' method that creates instances of both classes and calls their respective methods. The console output shows the results: 'Chef is cooking' and 'Cleaner is cleaning'.

```
1 package com.wipro.komal.solid_principal;
2
3 interface Cooking {
4     void cook();
5 }
6
7 interface Cleaning {
8     void clean();
9 }
10
11
12 class Chef implements Cooking {
13     public void cook() {
14         System.out.println("Chef is cooking");
15     }
16 }
17
18 class Cleaner implements Cleaning {
19     public void clean() {
20         System.out.println("Cleaner is cleaning");
21     }
22 }
23
24 public class ISPEExample {
25     public static void main(String[] args) {
26         Cooking chef = new Chef();
27         chef.cook();
28
29         Cleaning cleaner = new Cleaner();
30         cleaner.clean();
31     }
32 }
```

Console Output:

```
<terminated> ISPEExample [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.j
Chef is cooking
Cleaner is cleaning
```

5)



The screenshot shows the Eclipse IDE interface. The main editor displays the file `DIPEXample.java` with the following code:

```
1 package com.wipro.komal.solid_principal;
2
3 interface Engine {
4     void start();
5 }
6
7 class ElectricalEngine implements Engine {
8     @Override
9     public void start() {
10         System.out.println("Engine started");
11     }
12 }
13
14 class Car {
15     private Engine engine;
16
17     public Car(Engine engine) {
18         this.engine = engine;
19     }
20
21     void drive() {
22         engine.start();
23         System.out.println("Car started driving");
24     }
25 }
26
27 public class DIPEXample {
28     public static void main(String[] args) {
29         Engine engine = new ElectricalEngine();
30         Car car = new Car(engine);
31         car.drive();
32     }
33 }
```

The console window on the right shows the output of the program:

```
<terminated> DIPEXample [Java Application] C:\sts-4.31.0.RELEASE\plugins\org.eclipse.j
Engine started
Car started driving
```