Task 04:

class Employee {

    private String name;

    private String email;

    private double salary;

    // Methods related to employee data

    // Method to generate PDF report

    public void generatePdfReport() {

        // Code to generate PDF report

    }

    // Method to send email

    public void sendEmail() {

        // Code to send email

    }

}

In the above example code, the Employee class violates the SRP because it has multiple responsibilities: managing employee data, generating PDF reports, and sending emails. These responsibilities are not cohesive and may change for different reasons.

Code:

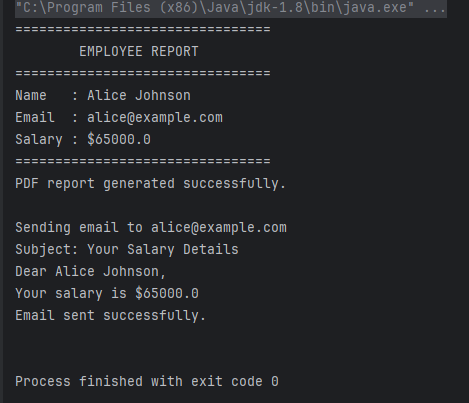
package July25\_email;  
  
public class EmailService {  
 public void sendEmail(Employee employee) {  
 // Simulate sending an email  
 System.*out*.println("Sending email to " + employee.getEmail());  
 System.*out*.println("Subject: Your Salary Details");  
 System.*out*.println("Dear " + employee.getName() + ",");  
 System.*out*.println("Your salary is $" + employee.getSalary());  
 System.*out*.println("Email sent successfully.\n");  
 }  
}

package July25\_email;  
  
public class Employee {  
 private String name;  
 private String email;  
 private double salary;  
  
 public Employee(String name, String email, double salary) {  
 this.name = name;  
 this.email = email;  
 this.salary = salary;  
 }  
  
 // Getters  
 public String getName() {  
 return name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 // Setters  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
}

package July25\_email;  
  
public class Main {  
 public static void main(String[] args) {  
 // Create an employee  
 Employee employee = new Employee("Alice Johnson", "alice@example.com", 65000.00);  
  
 // Generate PDF report  
 PdfReportGenerator pdfGenerator = new PdfReportGenerator();  
 pdfGenerator.generate(employee);  
  
 // Send email  
 EmailService emailService = new EmailService();  
 emailService.sendEmail(employee);  
 }  
}

package July25\_email;  
  
public class PdfReportGenerator {  
 public void generate(Employee employee) {  
 // Simulate PDF generation logic  
 System.*out*.println("================================");  
 System.*out*.println(" EMPLOYEE REPORT ");  
 System.*out*.println("================================");  
 System.*out*.println("Name : " + employee.getName());  
 System.*out*.println("Email : " + employee.getEmail());  
 System.*out*.println("Salary : $" + employee.getSalary());  
 System.*out*.println("================================");  
 System.*out*.println("PDF report generated successfully.\n");  
 }  
}

output:



Task 05:

Open closed Priciple  …

class Square() {

  int height;

  int area() { return height \* height; }

}

public class OpenOpenExample {

  public int compareArea(Square a, Square b) {

    return a.area() - b.area();

  }

}

extension code:

class Circle {

  int r;

  int area() { return Math.PI\*r\*r\*;}

}

class OpenOpenExample {

  public int compareArea(Square a, Square b) {

    return a.area() - b.area();

  }

  public int compareArea(Circle x, Circle y) {

   return x.area() - y.area();

  }

}

Code:

public interface Shape {

double area();

}

public class Square implements Shape {

private int height;

public Square(int height) {

this.height = height;

}

@Override

public double area() {

return height \* height;

}

}

public class Circle implements Shape {

private int r;

public Circle(int r) {

this.r = r;

}

@Override

public double area() {

return Math.PI \* r \* r;

}

}

public class AreaComparator {

public int compareArea(Shape a, Shape b) {

return Double.compare(a.area(), b.area());

}

}

public class Main {

public static void main(String[] args) {

Shape square1 = new Square(5);

Shape square2 = new Square(7);

Shape circle1 = new Circle(3);

Shape circle2 = new Circle(4);

AreaComparator comparator = new AreaComparator();

System.out.println("Square vs Square: " + comparator.compareArea(square1, square2));

System.out.println("Circle vs Circle: " + comparator.compareArea(circle1, circle2));

System.out.println("Square vs Circle: " + comparator.compareArea(square1, circle1));

}

}

output:

Square vs Square: -24

Circle vs Circle: -25

Square vs Circle: 3