**Task 1**

**Without SRP:**

SRP violation code

public class Customer {

String name;

String custID;

public Customer(String name, String custID) {

this.name = name;

this.custID = custID;

}

public String getName() {

return name;

}

public String getCustID {

return custID;

}

public void saveData() {

try{

FileWriter fw = new FileWriter(name+".txt");

fw.write("the customer name is "+ name + "\t");

fw.write("the customer id is "+ custID + "\t");

System.out.println("the data is saved in

the file with your name");

}catch(IOException ex) {

ex.printStacktrace();

}

}

psvm( String[] args) {

Customer cobj = new Customer(" prasunamba" , "C001");

cobj.saveData();

}

}

**SRP Implementation:**

Customer.java

public class Customer {

String name;

String custID;

public Customer(String name, String custID) {

this.name = name;

this.custID = custID;

}

public String getName() {

return name;

}

public String getCustID {

return custID;

}

}

ManagingFiles.java

import java.io.FileWriter;

import java.io.IOException;

public class ManagingFiles{

public void saveData() {

try{

FileWriter fw = new FileWriter(name+".txt");

fw.write("the customer name is "+ name + "\t");

fw.write("the customer id is "+ custID + "\t");

System.out.println("the data is saved in

the file with your name");

}catch(IOException ex) {

ex.printStacktrace();

}

}

}

SRP\_Imple.java

public class SRP\_Imple {

psvm( String[] args) {

Customer cobj = new Customer(" prasunamba" , "C001");

ManagingFiles mobj = new ManagingFiles();

mobj.saveData();

}

}

**Task 2**

**OCP -**

public class BankNotifications {

public void sendOTP(String via) {

if (via.equals("email")) {

sout("email sent to your mail id");

}

}

}

----in the above code if you want to send mail to your

mobile or whatsapp -- modify the above code or not?

**OCP - implementation**

public interface BankNotifications {

public void sendOTP(String via);

//public void TransactionNotification(Srting via);

//violates srp so .. include another interface

}

class EmailNotify implements BankNotifications {

public void sendOTP(String via) {

sout("email sent to your mail id");

}

/\*public void TransactionNotification(String via) {

sout("email sent to your mail id");

}\*/

}

class MobileNotify implements BankNotifications {

public void sendOTP(String via) {

sout("msg sent to your Mobile no");

}

/\*public void TransactionNotification(String via) {

sout("msg sent to your Mobile no");

}\*/

}

class WhatsappNotify implements BankNotifications {

public void sendOTP(String via) {

sout("msg sent to your whatsapp ");

}

/\*public void TransactionNotification(String via) {

sout("msg sent to your whatsapp");

}\*/

}

**Task 3**

**The below is violating SRP. Complete it and also implement the SRP principle and rewrite the code.**

// srp violation

public class Book {

private String title;

private String author;

private double price;

public Book(String title, String author, double price) {

this.title = title;

this.author = author;

this.price = price;

}

public String getFormattedTitle() {

return "Title: " + title.toUpperCase();

}

public double calculateDiscountedPrice(double discountPercentage) {

return price \* (1 - discountPercentage);

}

// ... other methods for book details

}

**Correcting it to maintain SRP**

public class BookDetails {

private String title;

private String author;

private double price;

public BookDetails(String title, String author, double price) {

this.title = title;

this.author = author;

this.price = price;

}

// ... getters and setters for book details

}

public class BookFormatter {

public String formatTitle(String title) {

return "Title: " + title.toUpperCase();

}

}

public class PriceCalculator {

public double calculateDiscountedPrice(double originalPrice, double discountPercentage) {

return originalPrice \* (1 - discountPercentage);

}

}

**Task 4**

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.

**Correcting it to maintain SRP**

class Employee

{

private String name;

private String email;

private double salary;

// Methods related to employee data

}

class GeneratePdf

{

// Method to generate PDF report

public void generatePdfReport()

{

// Code to generate PDF report

}

}

class ShareMail

{

// Method to send email

public void sendEmail()

{

// Code to send email

}

}

**Task 5**

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();

}

}

**Correcting it to maintain SRP**

interface Shape {

int area();

}

class Circle implements Shape {

int r;

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

}

class Square() implements Shape {

int height;

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

}

public class OpenClosedExample {

public int compareArea(Shape a, Shape b) {

return a.area() - b.area();

}

}

**Task 6**

**Can you guys create diagrams for structural diagrams…**

@startuml

class Laptop {

- int memory

- int storage

+ setMemory(int memory)

+ setStorage(int storage)

}

class LaptopBuilder {

+ buildMemory(int memory)

+ buildStorage(int storage)

+ build() : Laptop

}

class LaptopConcreteBuilder {

- Laptop laptop

+ buildMemory(int memory)

+ buildStorage(int storage)

+ build() : Laptop

}

LaptopConcreteBuilder ..|> LaptopBuilder

LaptopConcreteBuilder --> Laptop

@enduml

