WEEK 1:

MODULE 1 : - Design Patterns and Principles

**Exercise 1: Implementing the Singleton Pattern**

Code:  
Package DesignPatterns;  
class Singleton {

private static Singleton singleInstance;

private Singleton() {

System.out.println("Singleton Instance Created!");

}

public static Singleton getInstance() {

if (singleInstance == null) {

singleInstance = new Singleton();

}

return singleInstance;

}

public void showMessage() {

System.out.println("Hello from Singleton!");

}

}

package DesignPatterns;

public class SingletonDemo {

public static void main(String[] args) {

Singleton obj1 = Singleton.getInstance();

obj1.showMessage();

Singleton obj2 = Singleton.getInstance();

obj2.showMessage();

if (obj1 == obj2) {

System.out.println("Both objects are the same instance.");

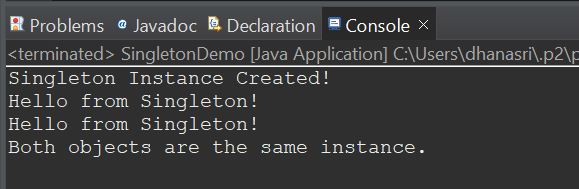
} else

System.out.println("Different instances exist!"); }

}

}

Output:



**Exercise 2: Implementing the Factory Method Pattern**

Code:

package DesignPatterns;

interface shape {

void draw();

}

package DesignPatterns;

class Circle implements shape {

public void draw() {

System.out.println("Drawing a Circle");

}

}

package DesignPatterns;

class rectangle implements shape {

public void draw() {

System.out.println("Drawing a Rectangle");

}

}

package DesignPatterns;

class ShapeFactory {

public shape getShape(String shapeType) {

if (shapeType == null) {

return null;

}

if(shapeType.equalsIgnoreCase("CIRCLE")) {

return new Circle();

} else if(shapeType.equalsIgnoreCase("RECTANGLE")) {

return new rectangle();

}

return null;

}

}

package DesignPatterns;

public class FactoryMethodDemo {

public static void main(String[] args) {

ShapeFactory shapeFactory = new ShapeFactory();

shape shape1 = shapeFactory.getShape("CIRCLE");

shape1.draw();

shape shape2 = shapeFactory.getShape("RECTANGLE");

shape2.draw();

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Module 2: Data structures and Algorithms

**Exercise 2: E-commerce Platform Search Function**

Code:

package Algorithms\_DataStructures;

class Product {

private int id;

private String name;

private double price;

public Product(int id, String name, double price) {

this.id = id;

this.name = name;

this.price = price;

}

public String getName() {

return name;

}

public void displayProduct() {

System.out.println("ID: " + id + ", Name: " + name + ", Price: $" + price);

}

}

package Algorithms\_DataStructures;

import java.util.ArrayList;

import java.util.List;

class ECommercePlatform {

private List<Product> products;

public ECommercePlatform() {

products = new ArrayList<>();

}

public void addProduct(Product product) {

products.add(product);

}

public void searchProduct(String searchQuery) {

System.out.println("Search Results for: " + searchQuery);

boolean found = false;

for (Product product : products) {

if (product.getName().toLowerCase().contains(searchQuery.toLowerCase())) {

product.displayProduct();

found = true;

}

}

if (!found) {

System.out.println("No products found matching the search query.");

}

}

}

package Algorithms\_DataStructures;

public class ECommerceSearchDemo {

public static void main(String[] args) {

ECommercePlatform platform = new ECommercePlatform();

platform.addProduct(new Product(1, "Laptop", 800.00));

platform.addProduct(new Product(2, "Smartphone", 500.00));

platform.addProduct(new Product(3, "Headphones", 50.00));

platform.addProduct(new Product(4, "Smartwatch", 200.00));

platform.searchProduct("smart");

platform.searchProduct("laptop");

platform.searchProduct("camera");

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 7: Financial Forecasting**

Code:

package Algorithms\_DataStructures;

class FinancialForecast {

public static double calculateFutureValue(double principal, double annualRate, int years, int compoundsPerYear) {

double amount = principal \* Math.pow((1 + (annualRate / compoundsPerYear)), (compoundsPerYear \* years));

return amount;

}

}

package Algorithms\_DataStructures;

public class FinancialForecastDemo {

public static void main(String[] args) {

double principal = 10000.0;

double annualRate = 0.08;

int years = 5;

int compoundsPerYear = 1;

double futureValue = FinancialForecast.calculateFutureValue(principal, annualRate, years, compoundsPerYear);

System.out.printf("Initial Investment: ₹%.2f%n", principal);

System.out.printf("Annual Interest Rate: %.2f%%%n", annualRate \* 100);

System.out.printf("Investment Period: %d years%n", years);

System.out.printf("Future Value: ₹%.2f%n", futureValue);

}

}

OUTPUT:

A screen shot of a computer

AI-generated content may be incorrect.