Task 5

E-Commerce System

Project Overview

Write a clear description of your project goals and objectives. For example:

Example:

Project Name: E-Commerce System with Product Hierarchy
Goal: To create a Java-based e-commerce system that allows users to browse products, add them to a shopping cart, and place orders.
Objectives:

- Implement an abstract Product class and concrete subclasses (ElectronicsProduct, ClothingProduct, BookProduct)
- Use inheritance and polymorphism to handle different product types
- Manage shopping cart and order functionality
- Apply discounts based on product type
- Provide a simple customer interaction system

Project Objectives

1. Implement Object-Oriented Programming Concepts
    - Utilize abstraction, inheritance, and polymorphism to create a flexible product hierarchy.
2. Design a Product Hierarchy
    - Create an abstract Product class with shared attributes and concrete subclasses for Electronics, Clothing, and Books.
3. Enable Shopping Cart Functionality ○ Allow users to add, remove, and update products in a shopping cart.
    - Calculate item totals and overall cart total dynamically.

4. Implement Order Management
    - Provide checkout functionality with order ID, date, subtotal, taxes, and final amount.
    - Generate a clear order summary for the customer.
5. Calculate Discounts and Taxes ○ Apply product-specific discounts.
    - Include tax calculation (e.g., GST) during checkout.
6. Support User Interaction ○ Provide a menu-driven interface to navigate products, cart, and checkout.
    - Handle invalid inputs and guide the user for correct actions.
7. Maintain Data Organization and Modularity
    - Structure code logically to allow easy extension (e.g., adding new product types in the future).
8. Demonstrate Practical Application
    - Simulate a real-world e-commerce experience for users, emphasizing cart and order management.

Setup & Installation Guide for IntelliJ IDEA

1. Install Java JDK

1. Download and install the latest Java JDK or OpenJDK.
2. Verify installation:
    - Open a terminal/command prompt and type:

```
java -version
```

2. Install IntelliJ IDEA

1. Download and install IntelliJ IDEA Community Edition

2. Launch IntelliJ IDEA after installation.

3. Create a New Java Project

1. Open IntelliJ → Click New Project.
2. Select Java → Click Next.
3. Set Project Name (e.g., ECommerceSystem) and choose a project location.
4. Make sure the correct JDK version is selected → Click Finish.

CODE STRUCTURE

```
import java.util.*;

// Abstract Product class
abstract class Product {
    protected String id;
    protected String name;
    protected double price;
    protected String description;

    public Product(String id, String name, double price, String description) {
        this.id = id;
        this.name = name;
        this.price = price;
        this.description = description;
    }

    public abstract double calculateDiscount();

    public double getFinalPrice() {
        return price - calculateDiscount();
    }

    public void displayInfo() {
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Price: ₹" + price);
        System.out.println("Description: " + description);
        System.out.println("Discount: ₹" + calculateDiscount());
        System.out.println("Final Price: ₹" + getFinalPrice());
        System.out.println("-".repeat(40));
    }

    public String getId() { return id; }
    public String getName() { return name; }
}

// Electronics product
class ElectronicsProduct extends Product {
    private String brand;
```

```
    public ElectronicsProduct(String id, String name, double price, String description, String
brand) {
        super(id, name, price, description);
        this.brand = brand;
    }

    @Override
    public double calculateDiscount() {
        return price * 0.10; // 10% discount
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Type: Electronics");
        System.out.println("Brand: " + brand);
        System.out.println("-".repeat(40));
    }
}

// Clothing product
class ClothingProduct extends Product {
    private String size;
    private String color;

    public ClothingProduct(String id, String name, double price, String description, String size,
String color) {
        super(id, name, price, description);
        this.size = size;
        this.color = color;
    }

    @Override
    public double calculateDiscount() {
        return price * 0.15; // 15% discount
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
```

```
        System.out.println("Type: Clothing");
        System.out.println("Size: " + size + ", Color: " + color);
        System.out.println("-".repeat(40));
    }
}

// Book product
class BookProduct extends Product {
    private String author;

    public BookProduct(String id, String name, double price, String description, String author) {
        super(id, name, price, description);
        this.author = author;
    }

    @Override
    public double calculateDiscount() {
        return price * 0.05; // 5% discount
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Type: Book");
        System.out.println("Author: " + author);
        System.out.println("-".repeat(40));
    }
}

// Cart item
class CartItem {
    private Product product;
    private int quantity;

    public CartItem(Product product, int quantity) {
        this.product = product;
        this.quantity = quantity;
    }

    public Product getProduct() { return product; }
    public int getQuantity() { return quantity; }
```

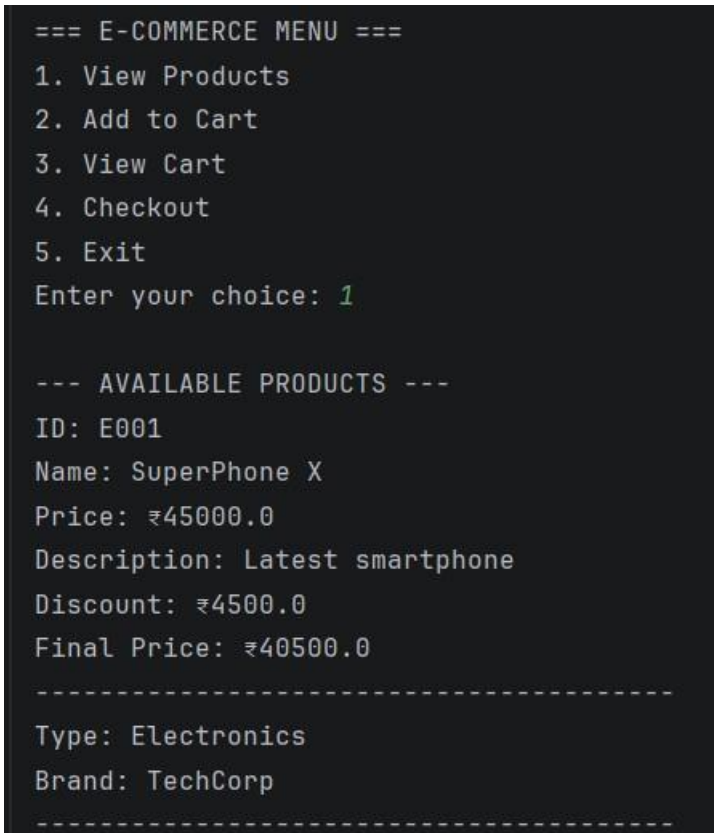
```
    public void setQuantity(int qty) { this.quantity = qty; }
    public double getItemTotal() { return product.getFinalPrice() * quantity; }
}

// Shopping cart
class ShoppingCart {
    private List<CartItem> items = new ArrayList<>();

    public void addItem(Product p, int qty) {
        for (CartItem ci : items) {
            if (ci.getProduct().getId().equals(p.getId())) {
                ci.setQuantity(ci.getQuantity() + qty);
                return;
            }
        }
        items.add(new CartItem(p, qty));
    }

    public void displayCart() {
        if (items.isEmpty()) {
            System.out.println("Cart is empty!");
            return;
        }
        System.out.println("=== SHOPPING CART ===");
        System.out.printf("%-10s %-20s %-10s %-5s %-10s\n", "ID", "Name", "Price", "Qty",
"Total");
        System.out.println("-".repeat(60));
        for (CartItem ci : items) {
            Product p = ci.getProduct();
            System.out.printf("%-10s %-20s ₹%-9.2f %-5d ₹%-9.2f\n",
                    p.getId(), p.getName(), p.getFinalPrice(), ci.getQuantity(), ci.getItemTotal());
        }
        System.out.println("-".repeat(60));
        System.out.printf("Total Amount: ₹%.2f\n", getTotalAmount());
    }

    public double getTotalAmount() {
        double total = 0;
        for (CartItem ci : items) total += ci.getItemTotal();
        return total;
    }
```

```
    public void clearCart() { items.clear(); }
}

// Order class
class Order {
    private static int counter = 1000;
    private String orderId;
    private double finalAmount;

    public Order(double amount) {
        this.orderId = "ORD" + (counter++);
        this.finalAmount = amount * 1.18; // Add 18% GST
    }

    public void displayOrder() {
        System.out.println("\n=== ORDER DETAILS ===");
        System.out.println("Order ID: " + orderId);
        System.out.printf("Final Amount (with GST): ₹%.2f\n", finalAmount);
        System.out.println("Thank you for your order!");
        System.out.println("-".repeat(40));
    }
}

// Main program
public class ECommerceSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Create your own products here
        List<Product> products = new ArrayList<>();
        products.add(new ElectronicsProduct("E001", "SuperPhone X", 45000, "Latest
smartphone", "TechCorp"));
        products.add(new ClothingProduct("C001", "Denim Jacket", 2500, "Stylish jacket", "L",
"Blue"));
        products.add(new BookProduct("B001", "Python Programming", 900, "Learn Python
easily", "Alice Smith"));

        ShoppingCart cart = new ShoppingCart();

        while (true) {
```

```
System.out.println("\n=== E-COMMERCE MENU ===");
System.out.println("1. View Products");
System.out.println("2. Add to Cart");
System.out.println("3. View Cart");
System.out.println("4. Checkout");
System.out.println("5. Exit");
System.out.print("Enter your choice: ");
int choice = sc.nextInt();
sc.nextLine(); // consume newline

switch (choice) {
    case 1:
        System.out.println("\n--- AVAILABLE PRODUCTS ---");
        for (Product p : products) p.displayInfo();
        break;
    case 2:
        System.out.print("Enter Product ID to add: ");
        String id = sc.nextLine();
        System.out.print("Enter Quantity: ");
        int qty = sc.nextInt();
        sc.nextLine();
        boolean found = false;
        for (Product p : products) {
            if (p.getId().equals(id)) {
                cart.addItem(p, qty);
                System.out.println("☑ " + p.getName() + " added to cart!");
                found = true;
                break;
            }
        }
        if (!found) System.out.println("Product not found!");
        break;
    case 3:
        cart.displayCart();
        break;
    case 4:
        cart.displayCart();
        Order order = new Order(cart.getTotalAmount());
        order.displayOrder();
        cart.clearCart();
        break;
```

```
            case 5:
                System.out.println("Exiting system. Goodbye!");
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice! Try again.");
        }
    }
  }
}
```

VISUAL DOCUMENTATION



```
ID: C001
Name: Denim Jacket
Price: ₹2500.0
Description: Stylish jacket
Discount: ₹375.0
Final Price: ₹2125.0
----------------------------------------
Type: Clothing
Size: L, Color: Blue
----------------------------------------
ID: B001
Name: Python Programming
Price: ₹900.0
Description: Learn Python easily
Discount: ₹45.0
Final Price: ₹855.0
----------------------------------------
Type: Book
Author: Alice Smith
----------------------------------------
```

```
=== E-COMMERCE MENU ===
1. View Products
2. Add to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 2
Enter Product ID to add: E101
Enter Quantity: 2
Product not found!

=== E-COMMERCE MENU ===
1. View Products
2. Add to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 3
Cart is empty!
```

```
=== E-COMMERCE MENU ===
1. View Products
2. Add to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 4
Cart is empty!

=== ORDER DETAILS ===
Order ID: ORD1000
Final Amount (with GST): ₹0.00
Thank you for your order!
----------------------------------------

=== E-COMMERCE MENU ===
1. View Products
2. Add to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 5
Exiting system. Goodbye!

Process finished with exit code 0
```

Technical Details

Explain **algorithms, data structures, and architecture**:

- **Inheritance & Polymorphism:**
    - Product is abstract. Each subclass overrides calculateDiscount() and displayInfo().
- **Data Structures:**
    - ArrayList to store products in the cart ○ for-each loops to calculate totals
- **Architecture:**
    - Menu-driven console application ○ Core classes: Product → CartItem → ShoppingCart → Order
- **Discount & Pricing Logic:**
    - Electronics → 10% discount ○ Clothing → 15% discount ○ Books → 5% discount ○ GST (tax) added at checkout

Testing Evidence

- **Test Cases:**

Test Case	Action	Expected Result	Pass/Fail
View Products	Choose option 1	List all products with details	Pass
Add to Cart	Add 2 units of E101	Cart shows correct total	Pass
Update Cart	Change quantity of C101 to 3	Cart total updated correctly	Pass
Checkout	Choose checkout	Order summary shows subtotal + tax	Pass