



RAJALAKSHMI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

ACADEMIC YEAR 2025 - 2026

SEMESTER III

OBJECT ORIENTED PROGRAMMING LABORATORY

MINI PROJECT REPORT

REGISTER NUMBER	2117240030165
NAME	YASHWANTH.M.C.
PROJECT TITLE	ONLINE ORDER MANAGEMENT SYSTEM
DATE OF SUBMISSION	
FACULTY IN-CHARGE	MRS STARLIN JENI

Signature of Faculty In-charge

Introduction:

An Online Order Management System is a Java-based mini project developed using Object-Oriented Programming concepts. The system helps users browse products, place orders, view order history, and process payments.

It demonstrates real-world usage of OOP concepts such as **Encapsulation, Abstraction, Inheritance, Polymorphism, Interfaces, Collections, and Exception Handling**.

This application also simulates a simple backend using Java collections and provides a basic console-based UI that mimics real-world e-commerce order management workflow.

Objectives:

The main objectives of this project are:

1. To implement a functional order management system using OOP principles.
2. To maintain product catalogs and user orders.
3. To simulate real-time operations like adding items to cart and checking out.
4. To design a structured menu-driven interface for easy interaction.
5. To demonstrate modular programming using Java classes and packages.
6. To introduce exception handling for smoother user operations.
7. To store and retrieve products using Java collections.

System Requirements and Setup:

Hardware Requirements:

- Processor: Intel i3 or above
- RAM: 4GB or above
- Minimum storage: 1GB

Software Requirements:

- OS: Windows / Linux / macOS
- JDK version: 8 or above
- Text editor / IDE:
- VS Code / IntelliJ IDEA / Eclipse / Notepad++

Setup:

1. Install JDK (Java Development Kit).
2. Configure Java PATH in environment variables.
3. Open the project folder in an IDE or terminal.
4. Compile using:

```
javac Main.java
```

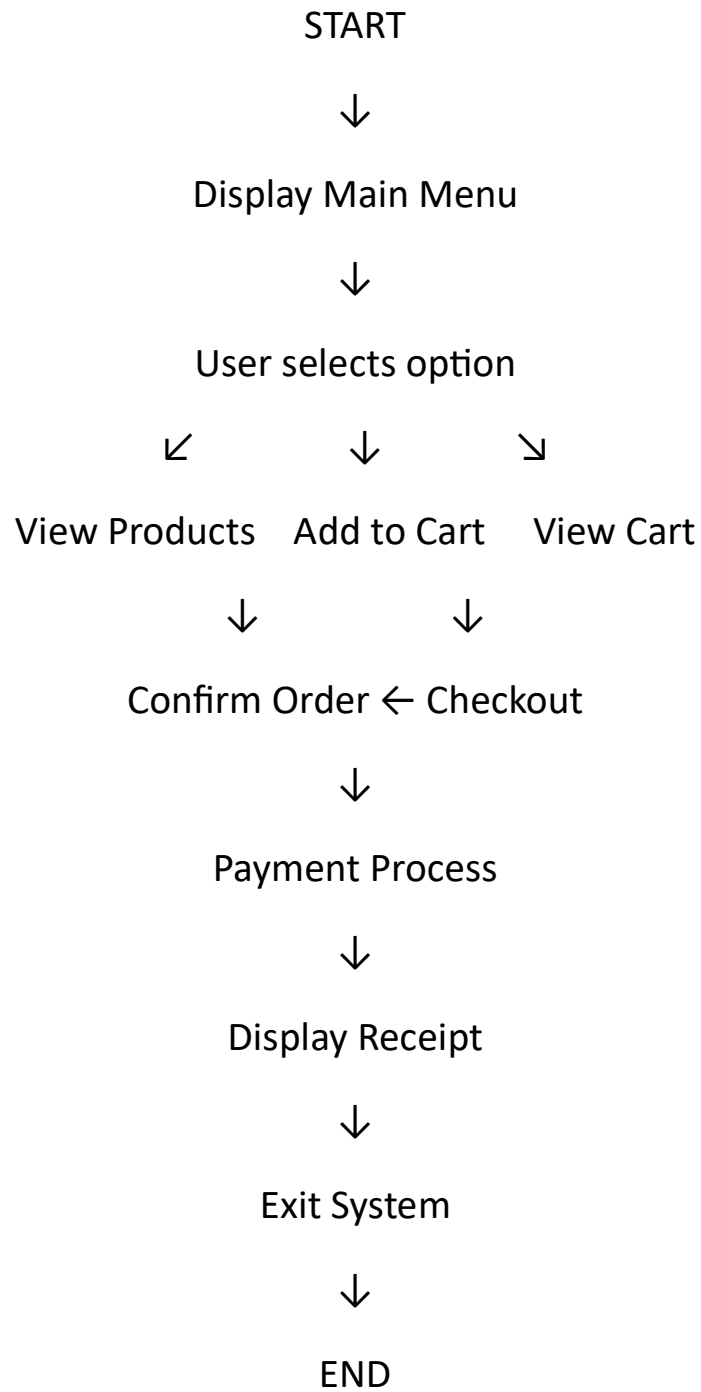
```
java Main
```

Procedure:

1. Identify the requirements for an online order system.
2. Design classes such as Product, Order, Cart, and User.
3. Create a product list using Java Collections.
4. Provide functionality to:
 - Display products
 - Add to cart
 - Remove from cart
 - Checkout
5. Implement OOP concepts:
 - Encapsulation (private data members)
 - Inheritance
 - Abstraction (interfaces)
 - Polymorphism
6. Create a console-based UI for user interaction.
7. Implement exception handling for invalid inputs.
8. Test the system with sample input.
9. Record results and generate screenshots.

Flow Diagram:

Below is a textual representation.



Code:

Complete Java Code (Console-based)

// Mini Project: Online Order Management System (GUI + File Backend)

// Subject: OOP - Java Programming

// Author: Yashwanth M.C.

// College: Rajalakshmi Institute of Technology

// Affiliated to Anna University, Chennai

```
import javax.swing.*;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
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 int getId() { return id; }
```

```
    public String getName() { return name; }
```

```
    public double getPrice() { return price; }
}

class Customer {
    private String name;
    private String email;
    public Customer(String name, String email) {
        this.name = name;
        this.email = email;
    }
    public String getName() { return name; }
    public String getEmail() { return email; }
}

class OrderItem {
    private Product product;
    private int quantity;
    public OrderItem(Product product, int quantity) {
        this.product = product;
        this.quantity = quantity;
    }
    public Product getProduct() { return product; }
    public int getQuantity() { return quantity; }
    public double getTotal() { return product.getPrice() * quantity; }
}

class Order {
```

```
private Customer customer;

private ArrayList<OrderItem> items = new ArrayList<>();

public Order(Customer customer) {
    this.customer = customer;
}

public void addItem(OrderItem item) {
    items.add(item);
}

public double getTotalAmount() {
    double total = 0;
    for (OrderItem item : items) {
        total += item.getTotal();
    }
    return total;
}

public void saveToFile() {
    try (FileWriter fw = new FileWriter("orders.txt", true)) {
        fw.write("Customer: " + customer.getName() + " (" + customer.getEmail() + ")\n");
        for (OrderItem item : items) {
            fw.write(item.getProduct().getName() + " x " + item.getQuantity() +
                "\n = ₹" + item.getTotal() + "\n");
        }
        fw.write("Total: ₹" + getTotalAmount() + "\n");
    }
}
```

```

        fw.write("-----\n");
    } catch (IOException e) {
        e.printStackTrace();
    }
}

}

public class OrderManagementApp extends JFrame {

    private ArrayList<Product> productList;

    private ArrayList<OrderItem> cart;

    private JTextField nameField, emailField, qtyField;

    private JTable table;

    private JTextArea summaryArea;

    public OrderManagementApp() {

        setTitle("Online Order Management System");

        setSize(750, 600);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

        setLocationRelativeTo(null);

        productList = new ArrayList<>();

        cart = new ArrayList<>();

        loadProducts();

        JPanel panel = new JPanel();

        panel.setLayout(new BorderLayout());

        // Customer details

        JPanel customerPanel = new JPanel(new GridLayout(2, 2, 5, 5));

```



```
customerPanel.setBorder(BorderFactory.createTitledBorder("Customer Details"));

customerPanel.add(new JLabel("Name:"));

nameField = new JTextField();

customerPanel.add(nameField);

customerPanel.add(new JLabel("Email:"));

emailField = new JTextField();

customerPanel.add(emailField);

// Product Table

String[] columnNames = {"Product ID", "Name", "Price (₹)"};

DefaultTableModel model = new DefaultTableModel(columnNames, 0);

for (Product p : productList) {

    model.addRow(new Object[]{p.getId(), p.getName(), p.getPrice()});

}

table = new JTable(model);

JScrollPane tableScroll = new JScrollPane(table);

tableScroll.setBorder(BorderFactory.createTitledBorder("Available Products"));

// Quantity + Add Button

JPanel addPanel = new JPanel();

addPanel.add(new JLabel("Quantity:"));

qtyField = new JTextField(5);

addPanel.add(qtyField);

JButton addBtn = new JButton("Add to Cart");

addPanel.add(addBtn);

// Order Summary Area
```

```
summaryArea = new JTextArea(10, 30);

summaryArea.setEditable(false);

JScrollPane summaryScroll = new JScrollPane(summaryArea);

summaryScroll.setBorder(BorderFactory.createTitledBorder("Order Summary"));

JButton orderBtn = new JButton("Place Order");

panel.add(customerPanel, BorderLayout.NORTH);

panel.add(tableScroll, BorderLayout.CENTER);

panel.add(addPanel, BorderLayout.SOUTH);

add(panel, BorderLayout.NORTH);

add(summaryScroll, BorderLayout.CENTER);

add(orderBtn, BorderLayout.SOUTH);

// Action Listeners

addBtn.addActionListener(e -> addToCart());

orderBtn.addActionListener(e -> placeOrder());

setVisible(true);

}

private void loadProducts() {

    productList.add(new Product(1, "Laptop", 55000));

    productList.add(new Product(2, "Smartphone", 25000));

    productList.add(new Product(3, "Headphones", 3000));

    productList.add(new Product(4, "Keyboard", 1500));

    productList.add(new Product(5, "Smartwatch", 7000));

}

private void addToCart() {
```

```
int row = table.getSelectedRow();

if (row == -1) {
    JOptionPane.showMessageDialog(this, "Select a product first!");
    return;
}

try {
    int qty = Integer.parseInt(qtyField.getText());

    if (qty <= 0) throw new NumberFormatException();

    Product selectedProduct = productList.get(row);
    cart.add(new OrderItem(selectedProduct, qty));

    summaryArea.append(selectedProduct.getName() + " x " + qty +
        " = ₹" + (selectedProduct.getPrice() * qty) + "\n");

    qtyField.setText("");
} catch (NumberFormatException ex) {
    JOptionPane.showMessageDialog(this, "Enter a valid quantity!");
}

}

private void placeOrder() {
    if (nameField.getText().isEmpty() || emailField.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Enter customer details!");
        return;
    }

    if (cart.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Cart is empty!");
    }
}
```

```
        return;
    }

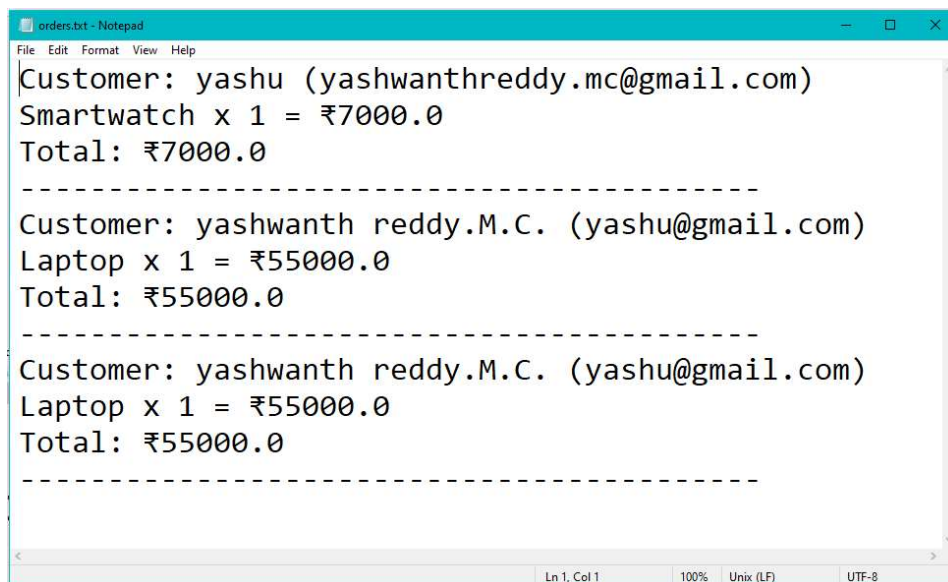
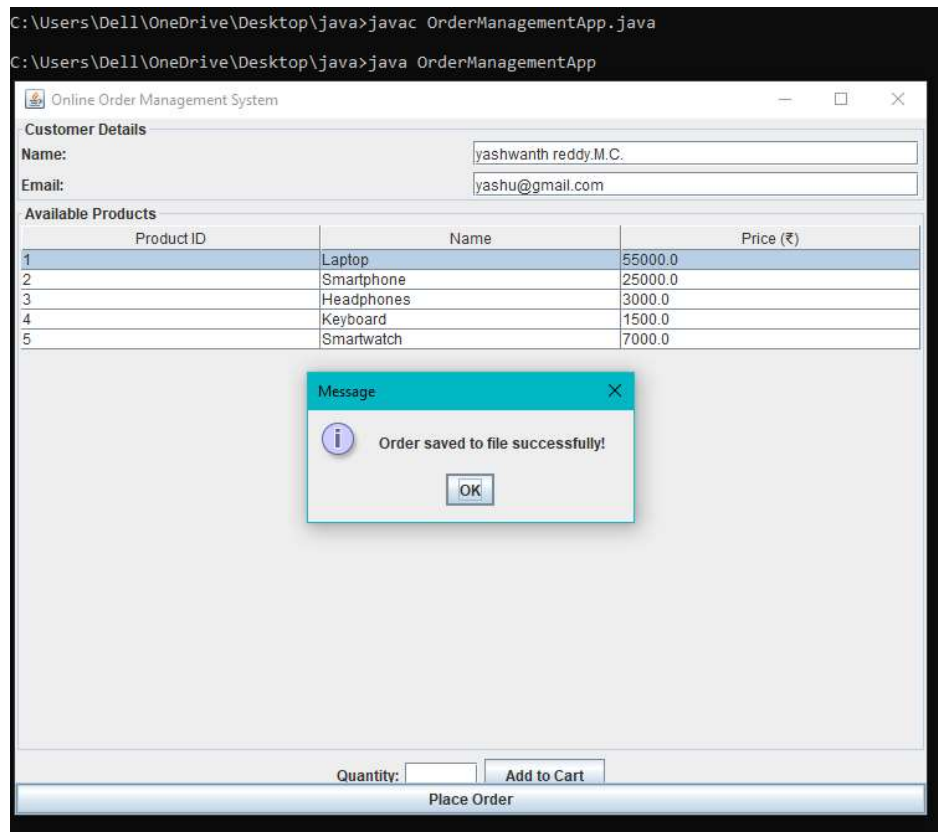
    Customer c = new Customer(nameField.getText(), emailField.getText());
    Order order = new Order(c);
    for (OrderItem item : cart) order.addItem(item);
    order.saveToFile();

    summaryArea.append("\n-----\n");
    summaryArea.append("Total: ₹" + order.getTotalAmount() + "\n");
    summaryArea.append("Order placed successfully!\n");
    JOptionPane.showMessageDialog(this, "Order saved to file successfully!");

    cart.clear();
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(OrderManagementApp::new);
}
}
```

OUTPUT:



GitHub Reference: <https://github.com/Yashwahthmc/oops-mini-project.git>