**COMSATS** **University Islamabad, Lahore Campus**

**Department of Computer Science**



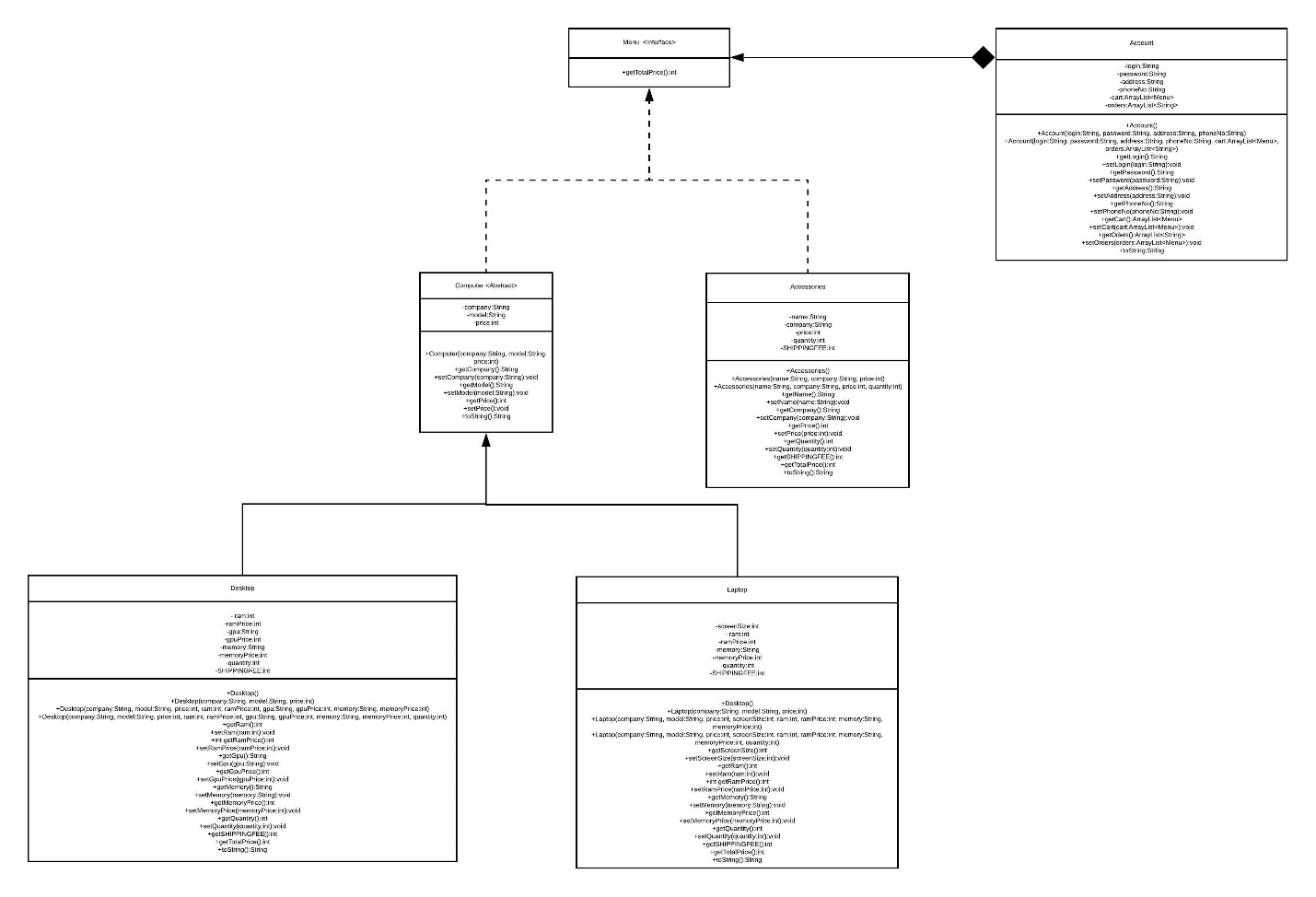
**Assignment 2 & 3 – Semester Spring 2020**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Title: | Object Oriented Programming-Lab | | | | | Course Code: | | CSC241 | Credit Hours: | 4(3,1) |
| Course Instructor/s: | Ms. Muntaha Iqbal | | | | | Programme Name: | | BCS | | |
| Semester: | 3rd | Batch: | SP19-BSCS | | Session: | C  Time  60 Minutes | | | | |
| Total Marks: | **40** Obtained Marks: | Obtained Marks: | |  | | Date: | August 03, 2020 | | | |
| Student’s Name: | **S.M. Haseeb Iqbal** | | | | | Reg. No. | SP19-BCS-042 | | | |
|  | | | | | | | | | | |

**Project Idea**

This application is an idea of online shopping store for Computers and their Accessories. It provides the user with a secure interface of Signup or login. User can select the desired object and can add it to their cart or can buy the item directly from the item menu. Moreover, user can check the description of the object also he/she can manage their cart. User can also check the list of their confirmed orders whenever they want.

**UML Diagram**

****

**Menu 🡪 Tagging interface**

package com.C\_Mart;  
  
import java.io.Serializable;  
  
public interface Menu extends Serializable {  
 public int getTotalPrice();  
}

**Computer 🡪Abstract Class**

package com.C\_Mart;  
  
public abstract class Computer implements Menu{  
 private String company;  
 private String model;  
 private int price;  
  
 public Computer(String company, String model, int price) {  
 this.company = company;  
 this.model = model;  
 this.price = price;  
 }  
  
 public String getCompany() {  
 return company;  
 }  
  
 public void setCompany(String company) {  
 this.company = company;  
 }  
  
 public String getModel() {  
 return model;  
 }  
  
 public void setModel(String model) {  
 this.model = model;  
 }  
  
 public int getPrice() {  
 return price;  
 }  
  
 public void setPrice(int price) {  
 this.price = price;  
 }  
  
 @Override  
 public String toString() {  
 return "Computer{" +  
 "company='" + company + '\'' +  
 ", model='" + model + '\'' +  
 ", price=" + price +  
 '}';  
 }  
}

**Desktop**

package com.C\_Mart;  
  
public class Desktop extends Computer{  
 private int ram;  
 private int ramPrice;  
 private String gpu;  
 private int gpuPrice;  
 private String memory;  
 private int memoryPrice;  
 private int quantity = 1;  
 private final int SHIPPINGFEE=1000;  
  
 public Desktop(String company, String model, int price) {  
 super(company, model, price);  
 }  
  
 public Desktop() {  
 super("", "", 0);  
 }  
  
 public Desktop(String company, String model, int price, int ram, int ramPrice, String gpu, int gpuPrice, String memory, int memoryPrice) {  
 super(company, model, price);  
 this.ram = ram;  
 this.ramPrice = ramPrice;  
 this.gpu = gpu;  
 this.gpuPrice = gpuPrice;  
 this.memory = memory;  
 this.memoryPrice = memoryPrice;  
 }  
  
 public Desktop(String company, String model, int price, int ram, int ramPrice, String gpu, int gpuPrice, String memory, int memoryPrice, int quantity) {  
 super(company, model, price);  
 this.ram = ram;  
 this.ramPrice = ramPrice;  
 this.gpu = gpu;  
 this.gpuPrice = gpuPrice;  
 this.memory = memory;  
 this.memoryPrice = memoryPrice;  
 this.quantity = quantity;  
 }  
  
 public int getRam() {  
 return ram;  
 }  
  
 public void setRam(int ram) {  
 this.ram = ram;  
 }  
  
 public int getRamPrice() {  
 return ramPrice;  
 }  
  
 public void setRamPrice(int ramPrice) {  
 this.ramPrice = ramPrice;  
 }  
  
 public String getGpu() {  
 return gpu;  
 }  
  
 public void setGpu(String gpu) {  
 this.gpu = gpu;  
 }  
  
 public int getGpuPrice() {  
 return gpuPrice;  
 }  
  
 public void setGpuPrice(int gpuPrice) {  
 this.gpuPrice = gpuPrice;  
 }  
  
 public String getMemory() {  
 return memory;  
 }  
  
 public void setMemory(String memory) {  
 this.memory = memory;  
 }  
  
 public int getMemoryPrice() {  
 return memoryPrice;  
 }  
  
 public void setMemoryPrice(int memoryPrice) {  
 this.memoryPrice = memoryPrice;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public void setQuantity(int quantity) {  
 this.quantity = quantity;  
 }  
  
 public int getSHIPPINGFEE() {  
 return SHIPPINGFEE;  
 }  
  
 @Override  
 public int getTotalPrice() {  
 return (getQuantity()\*(getMemoryPrice()+getPrice()+getGpuPrice()))+SHIPPINGFEE;  
 }  
  
 @Override  
 public String toString() {  
 return "---------Desktop---------" +  
 "\nCompany => "+super.getCompany()+"\nModel => "+super.getModel()  
 +"\nPrice => " +super.getPrice()+  
 " Ram => " + ram +" GB Price: "+ ramPrice +  
 " \nGPU => " + gpu +" GPU Price=" + gpuPrice +  
 "\nMemory => " + memory + " MemoryPrice=" + memoryPrice +  
 " \nQuantity => " + quantity +  
 " \nSHIPPINGFEE=" + SHIPPINGFEE +  
 "\nTotal Price => "+getTotalPrice();  
 }  
}

**Laptop**

package com.C\_Mart;  
  
public class Laptop extends Computer{  
 private int screenSize;  
 private int ram;  
 private int ramPrice;  
 private String memory;  
 private int memoryPrice;  
 private int quantity = 1;  
 private final int SHIPPINGFEE = 500;  
  
 public Laptop(String company, String model, int price) {  
 super(company, model, price);  
 }  
  
 public Laptop(String company, String model, int price, int screenSize, int ram, int ramPrice, String memory, int memoryPrice) {  
 super(company, model, price);  
 this.screenSize = screenSize;  
 this.ram = ram;  
 this.ramPrice = ramPrice;  
 this.memory = memory;  
 this.memoryPrice = memoryPrice;  
 }  
  
 public Laptop(String company, String model, int price, int screenSize, int ram, int ramPrice, String memory, int memoryPrice, int quantity) {  
 super(company, model, price);  
 this.screenSize = screenSize;  
 this.ram = ram;  
 this.ramPrice = ramPrice;  
 this.memory = memory;  
 this.memoryPrice = memoryPrice;  
 this.quantity = quantity;  
 }  
  
 public int getScreenSize() {  
 return screenSize;  
 }  
  
 public void setScreenSize(int screenSize) {  
 this.screenSize = screenSize;  
 }  
  
 public int getRam() {  
 return ram;  
 }  
  
 public void setRam(int ram) {  
 this.ram = ram;  
 }  
  
 public int getRamPrice() {  
 return ramPrice;  
 }  
  
 public void setRamPrice(int ramPrice) {  
 this.ramPrice = ramPrice;  
 }  
  
 public String getMemory() {  
 return memory;  
 }  
  
 public void setMemory(String memory) {  
 this.memory = memory;  
 }  
  
 public int getMemoryPrice() {  
 return memoryPrice;  
 }  
  
 public void setMemoryPrice(int memoryPrice) {  
 this.memoryPrice = memoryPrice;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public void setQuantity(int quantity) {  
 this.quantity = quantity;  
 }  
  
 public int getSHIPPINGFEE() {  
 return SHIPPINGFEE;  
 }  
  
 @Override  
 public int getTotalPrice() {  
 return (getQuantity()\*(getPrice()+getRamPrice()+getMemoryPrice()))+SHIPPINGFEE;  
 }  
  
 @Override  
 public String toString() {  
 return "---------Laptop---------" +  
 "\nCompany => "+super.getCompany()+"\nModel => "+super.getModel()+" Screen Size => "+getScreenSize()  
 +"\nPrice => " +super.getPrice()+  
 " Ram => " + ram +" GB Price: "+ ramPrice +  
 "\nMemory => " + memory + " MemoryPrice=" + memoryPrice +  
 " \nQuantity => " + quantity +  
 " \nSHIPPINGFEE=" + SHIPPINGFEE +  
 "\nTotal Price => "+getTotalPrice();  
 }  
}

**Accessories**

package com.C\_Mart;  
  
public class Accessories implements Menu{  
 private String name;  
 private String company;  
 private int price;  
 private int quantity = 1;  
 private final int SHIPPINGFEE = 100;  
  
 public Accessories() {  
 }  
  
 public Accessories(String name, String company, int price) {  
 this.name = name;  
 this.company = company;  
 this.price = price;  
 }  
  
 public Accessories(String name, String company, int price, int quantity) {  
 this.name = name;  
 this.company = company;  
 this.price = price;  
 this.quantity = quantity;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getCompany() {  
 return company;  
 }  
  
 public void setCompany(String company) {  
 this.company = company;  
 }  
  
 public int getPrice() {  
 return price;  
 }  
  
 public void setPrice(int price) {  
 this.price = price;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public void setQuantity(int quantity) {  
 this.quantity = quantity;  
 }  
  
 public int getSHIPPINGFEE() {  
 return SHIPPINGFEE;  
 }  
  
 @Override  
 public int getTotalPrice() {  
 return (getQuantity()\*getPrice())+SHIPPINGFEE;  
 }  
  
 @Override  
 public String toString() {  
 return "---------Accessories---------" +  
 "\nName => " + name +"\nCompany => " + company +  
 "\nPrice=" + price +  
 "\nQuantity=" + quantity +  
 "\nSHIPPINGFEE=" + SHIPPINGFEE +  
 "\nTotal Price => "+getTotalPrice();  
 }  
}

**Account**

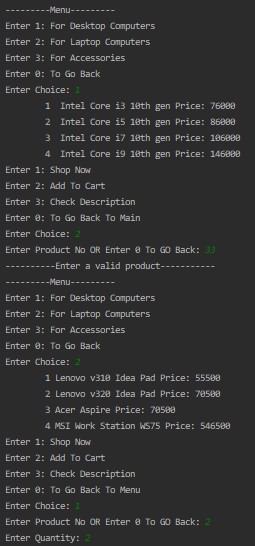
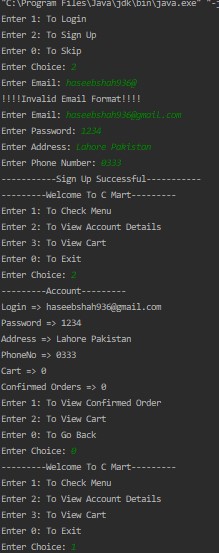
package com.C\_Mart;  
  
import java.io.Serializable;  
import java.util.ArrayList;  
  
public class Account implements Serializable {  
 private String login;  
 private String password;  
 private String address;  
 private String phoneNo;  
 private ArrayList<Menu> cart = new ArrayList<>();  
 private ArrayList<String> orders = new ArrayList<>();  
  
 public Account() {  
 }  
  
 public Account(String login, String password, String address, String phoneNo) {  
 this.login = login;  
 this.password = password;  
 this.address = address;  
 this.phoneNo = phoneNo;  
 }  
  
 public Account(String login, String password, String address, String phoneNo, ArrayList<Menu> cart, ArrayList<String> orders) {  
 this.login = login;  
 this.password = password;  
 this.address = address;  
 this.phoneNo = phoneNo;  
 this.cart = cart;  
 this.orders = orders;  
 }  
  
 public String getLogin() {  
 return login;  
 }  
  
 public void setLogin(String login) {  
 this.login = login;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 public String getAddress() {  
 return address;  
 }  
  
 public void setAddress(String address) {  
 this.address = address;  
 }  
  
 public String getPhoneNo() {  
 return phoneNo;  
 }  
  
 public void setPhoneNo(String phoneNo) {  
 this.phoneNo = phoneNo;  
 }  
  
 public ArrayList<Menu> getCart() {  
 return cart;  
 }  
  
 public void setCart(ArrayList<Menu> cart) {  
 this.cart = cart;  
 }  
  
 public ArrayList<String> getOrders() {  
 return orders;  
 }  
  
 public void setOrders(ArrayList<String> orders) {  
 this.orders = orders;  
 }  
  
 @Override  
 public String toString() {  
 return "---------Account---------" +  
 "\nLogin => " + login +  
 "\nPassword => " + password +  
 "\nAddress => " + address +  
 "\nPhoneNo => " + phoneNo +  
 "\nCart => " + cart.size() +  
 "\nConfirmed Orders => " + orders.size();  
 }  
}

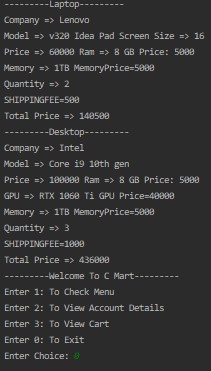
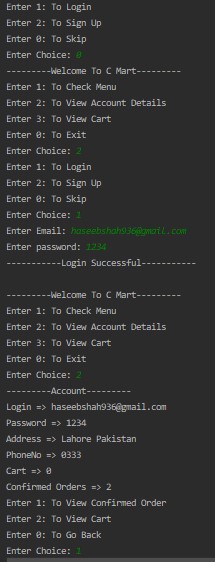
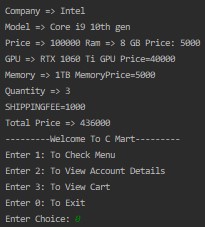
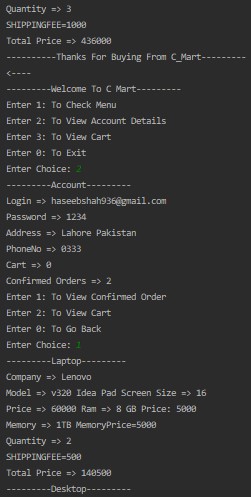
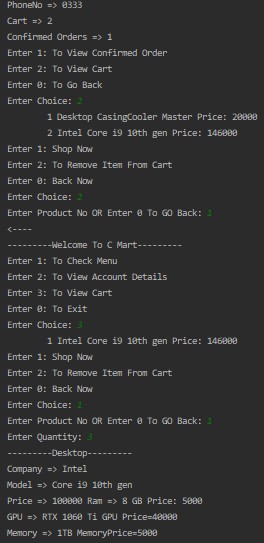
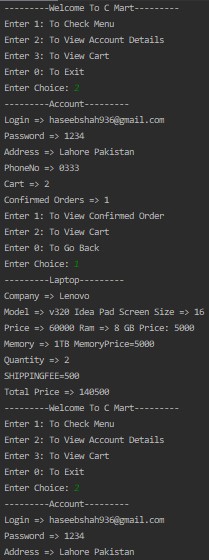
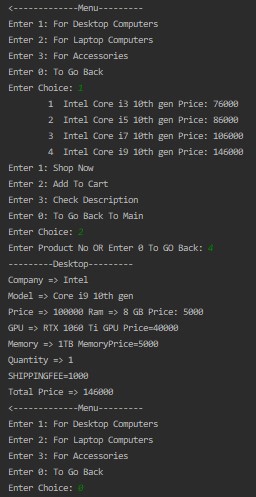
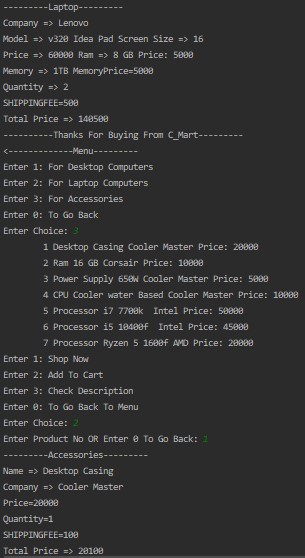
**Main 🡪 Test**

package com.C\_Mart;  
  
import java.io.\*;  
import java.lang.annotation.AnnotationTypeMismatchException;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.util.ArrayList;  
import java.util.NoSuchElementException;  
import java.util.Scanner;  
  
public class Main {  
 private static Account *account* = new Account();  
 private static Menu *menu*;  
 private static Scanner *scanner* = new Scanner(System.*in*);  
 private static int *Choice*;  
 private static int *condition* = 0;  
 private static ArrayList<String> *orders*;  
 private static ArrayList<Menu> *cart* ;  
  
 public static void main(String[] args) {  
 do {  
 try {  
 *Login*();  
 } catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 continue;  
 }  
 catch (IndexOutOfBoundsException e) {  
 System.*out*.println("----------Enter a valid product-----------");  
 continue;  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 continue;  
 }  
 break;  
 }while (true);  
  
 do{  
 try {  
 *condition* = *Interface*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 continue;  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 continue;  
 }  
  
 }while (*condition* == 0);  
 *writeAccount*();  
 }  
 private static int Interface() {  
  
 int inner\_condition = 0; // Menu Loop Terminator  
 System.*out*.println("---------Welcome To C Mart---------");  
 System.*out*.println("Enter 1: To Check Menu \nEnter 2: To View Account Details\nEnter 3: To View Cart\nEnter 0: To Exit");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1){  
 do {  
 try {  
 inner\_condition = *Menu*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 continue;  
 }  
 catch (IndexOutOfBoundsException e) {  
 System.*out*.println("----------Enter a valid product-----------");  
 continue;  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 continue;  
 }  
 }while (inner\_condition == 0);  
 }  
 else if(*Choice* == 2){  
 try {  
 *Account*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (IndexOutOfBoundsException e) {  
 System.*out*.println("----------Enter a valid product-----------");  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 else if(*Choice* == 3){  
 if(*account*.getLogin() == null){ //Checker For Account Login  
 try {  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 else{  
 try {  
 *Cart*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (IndexOutOfBoundsException e) {  
 System.*out*.println("----------Enter a valid product-----------");  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
  
 }  
 }  
 }  
 else{  
 return -1;  
 }  
 *scanner*.nextLine();  
 return 0;  
 }  
 private static int Menu(){ // Displaying Products for Shopping or Adding to Cart  
 int counter = 0;  
 int productNo = 0;  
 int quantity = 1;  
 *cart* = *account*.getCart();  
 *orders* = *account*.getOrders();  
 System.*out*.println("---------Menu---------");  
 System.*out*.println("Enter 1: For Desktop Computers \nEnter 2: For Laptop Computers\nEnter 3: For Accessories\nEnter 0: To Go Back");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1){  
 ArrayList<Menu> desktop = new ArrayList<>();  
 desktop.add(new Desktop("Intel","Core i3 10th gen",30000,8,5000,"RTX 1060 Ti",40000,"1TB",5000));  
 desktop.add(new Desktop("Intel","Core i5 10th gen",40000,8,5000,"RTX 1060 Ti",40000,"1TB",5000));  
 desktop.add(new Desktop("Intel","Core i7 10th gen",60000,8,5000,"RTX 1060 Ti",40000,"1TB",5000));  
 desktop.add(new Desktop("Intel","Core i9 10th gen",100000,8,5000,"RTX 1060 Ti",40000,"1TB",5000));  
  
 for (Menu m: desktop) {  
 Desktop d = (Desktop) m;  
 System.*out*.println(" "+(++counter)+" "+d.getCompany()+" "+d.getModel()+" Price: "+d.getTotalPrice());  
 }  
 System.*out*.println("Enter 1: Shop Now \nEnter 2: Add To Cart\nEnter 3: Check Description\nEnter 0: To Go Back To Main");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1){ // For Shop Now  
 if(*account*.getLogin() != null){ // Checker For Account Existence  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if(productNo == 0){  
  
 }  
 else{  
  
 Desktop order = (Desktop) desktop.get(productNo-1);  
 System.*out*.print("Enter Quantity: ");  
 quantity = *scanner*.nextInt();  
 order.setQuantity(quantity);  
 System.*out*.println(order);  
 *orders*.add(order.toString());  
 *account*.setOrders(*orders*);  
 System.*out*.println("----------Thanks For Buying From C\_Mart---------");  
 *writeAccount*();  
 }  
 }  
 else{  
 *scanner*.nextLine();  
 try {  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 System.*out*.print("<----");  
 *scanner*.nextLine();  
 }  
 else if(*Choice* == 2){ // For adding product to Cart  
 if(*account*.getLogin() != null) { // Checker For Account Existence  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) {  
  
 } else {  
 Desktop order = (Desktop) desktop.get(productNo - 1);  
 System.*out*.println(order);  
 *cart*.add(order);  
 *account*.setCart(*cart*);  
 *writeAccount*();  
 }  
 }  
 else{  
 *scanner*.nextLine();  
 try {  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 }  
 else if(*Choice* == 3){  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) {  
  
 } else {  
 Desktop order = (Desktop) desktop.get(productNo - 1);  
 System.*out*.println(order);  
 }  
 }  
 System.*out*.print("<----");  
 *scanner*.nextLine();  
 }  
 else if(*Choice* == 2){  
 ArrayList<Menu> laptop = new ArrayList<>();  
 laptop.add(new Laptop("Lenovo","v310 Idea Pad",45000,16,8,5000,"1TB",5000,1));  
 laptop.add(new Laptop("Lenovo","v320 Idea Pad",60000,16,8,5000,"1TB",5000,1));  
 laptop.add(new Laptop("Acer","Aspire",60000,16,8,5000,"1TB",5000,1));  
 laptop.add(new Laptop("MSI","Work Station WS75",531000,17,16,10000,"1TB",5000,1));  
  
 for (Menu m : laptop) {  
 Laptop d = (Laptop) m;  
 System.*out*.println(" "+(++counter)+" "+d.getCompany()+" "+d.getModel()+" Price: "+d.getTotalPrice());  
 }  
 System.*out*.println("Enter 1: Shop Now \nEnter 2: Add To Cart\nEnter 3: Check Description\nEnter 0: To Go Back To Menu");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1){ // For Shop Now  
 if(*account*.getLogin() != null) { // Checker For Account Existence  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) { // For Confirmation of Order and to Specify the product No.  
  
 } else {  
  
 Laptop order = (Laptop) laptop.get(productNo - 1);  
 System.*out*.print("Enter Quantity: ");  
 quantity = *scanner*.nextInt();  
 order.setQuantity(quantity);  
 System.*out*.println(order);  
 *orders*.add(order.toString());  
 *account*.setOrders(*orders*);  
 System.*out*.println("----------Thanks For Buying From C\_Mart---------");  
 *writeAccount*();  
 }  
 }  
 else {  
 *scanner*.nextLine();  
 try {  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 }  
 else if(*Choice* == 2){ // For adding product to Cart  
 if(*account*.getLogin() != null) { // Checker For Account Existence  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) {  
  
 } else {  
 Laptop order = (Laptop) laptop.get(productNo - 1);  
 System.*out*.println(order);  
 *cart*.add(order);  
 *account*.setCart(*cart*);  
 *writeAccount*();  
 }  
 }  
 else {  
 *scanner*.nextLine();  
 try {  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 }  
 else if(*Choice* == 3){  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) {  
  
 } else {  
 Laptop order = (Laptop) laptop.get(productNo - 1);  
 System.*out*.println(order);  
 }  
 }  
 System.*out*.print("<----");  
 *scanner*.nextLine();  
  
 }  
 else if(*Choice* == 3){  
 ArrayList<Menu> accessories = new ArrayList<>();  
 accessories.add(new Accessories("Desktop Casing","Cooler Master",20000));  
 accessories.add(new Accessories("Ram 16 GB","Corsair",10000));  
 accessories.add(new Accessories("Power Supply 650W","Cooler Master",5000));  
 accessories.add(new Accessories("CPU Cooler water Based","Cooler Master",10000));  
 accessories.add(new Accessories("Processor i7 7700k ","Intel",50000));  
 accessories.add(new Accessories("Processor i5 10400f ","Intel",45000));  
 accessories.add(new Accessories("Processor Ryzen 5 1600f","AMD",20000));  
  
 for (Menu m : accessories) {  
 Accessories d = (Accessories) m;  
 System.*out*.println(" "+(++counter)+" "+d.getName()+" "+d.getCompany()+" Price: "+d.getPrice());  
 }  
 System.*out*.println("Enter 1: Shop Now \nEnter 2: Add To Cart\nEnter 3: Check Description\nEnter 0: To Go Back To Menu");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1){ // For Shop Know  
 if(*account*.getLogin() != null) { // Checker For Account Existence  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) { // For Confirmation of Order and to Specify the product No.  
  
 } else {  
  
 Accessories order = (Accessories) accessories.get(productNo - 1);  
 System.*out*.print("Enter Quantity: ");  
 quantity = *scanner*.nextInt();  
 order.setQuantity(quantity);  
 System.*out*.println(order);  
 *orders*.add(order.toString());  
 *account*.setOrders(*orders*);  
 System.*out*.println("----------Thanks For Buying From C\_Mart---------");  
 *writeAccount*();  
 }  
 }  
 else {  
 *scanner*.nextLine();  
 try {  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 }  
 else if(*Choice* == 2){ // For adding product to Cart  
 if(*account*.getLogin() != null) { // Checker For Account Existence  
 System.*out*.print("Enter Product No OR Enter 0 To Go Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) {  
  
 } else {  
 Accessories order = (Accessories) accessories.get(productNo - 1);  
 System.*out*.println(order);  
 *cart*.add(order);  
 *account*.setCart(*cart*);  
 *writeAccount*();  
 }  
 }  
 else {  
 *scanner*.nextLine();  
 try {  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 }  
 else if(*Choice* == 3){  
 System.*out*.print("Enter Product No OR Enter 0 To Go Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) {  
  
 } else {  
 Accessories order = (Accessories) accessories.get(productNo - 1);  
 System.*out*.println(order);  
 }  
 }  
 System.*out*.print("<----");  
 *scanner*.nextLine();  
 }  
 else {  
 return -1;  
 }  
 return 0;  
 }  
 private static void Account() {  
 if(*account*.getLogin() == null){ //Checker For Account Login  
 try{  
 *Login*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
 }  
 }  
 else{  
 System.*out*.println(*account*);  
 System.*out*.println("Enter 1: To View Confirmed Order\nEnter 2: To View Cart\nEnter 0: To Go Back");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1){  
 for (String s : *account*.getOrders()) {  
 System.*out*.println(s);  
 }  
 }  
 else if(*Choice* == 2){  
 try {  
 *Cart*();  
 }  
 catch (NoSuchElementException e) {  
 System.*out*.println("----------Enter a Valid Number-----------");  
 *scanner*.nextLine();  
 }  
 catch (ArrayIndexOutOfBoundsException e) {  
 System.*out*.println("----------Enter a valid product-----------");  
 }  
 catch (Exception e) {  
 System.*out*.println("----------Enter a Valid Option-----------");  
  
 }  
 }  
 }  
 }  
 private static void Cart() {  
 *cart* = *account*.getCart();  
 *orders* = *account*.getOrders();  
 int counter = 0;  
 int productNo = 0;  
 int quantity = 1;  
 for (Menu m : *cart*) {  
 if (m instanceof Desktop) {  
 Desktop d = (Desktop) m;  
 System.*out*.println(" " + (++counter) + " " + d.getCompany() + " " + d.getModel() + " Price: " + d.getTotalPrice());  
 } else if (m instanceof Laptop) {  
 Laptop d = (Laptop) m;  
 System.*out*.println(" " + (++counter) + " " + d.getCompany() + " " + d.getModel() + " Price: " + d.getTotalPrice());  
 } else if (m instanceof Accessories) {  
 Accessories d = (Accessories) m;  
 System.*out*.println(" " + (++counter) + " " + d.getName() + d.getCompany() +" Price: "+ d.getPrice());  
 }  
 }  
 System.*out*.println("Enter 1: Shop Now\nEnter 2: To Remove Item From Cart\nEnter 0: Back Now");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1) {  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) { // For Confirmation of Order and to Specify the product No.  
  
 }  
 else { // After Confirmation of order adding order to confirmed order and removing from cart  
  
 Menu m = *cart*.get(productNo-1);  
 *cart*.remove(productNo-1);  
 if(m instanceof Desktop){  
 Desktop order = (Desktop) m;  
 System.*out*.print("Enter Quantity: ");  
 quantity = *scanner*.nextInt();  
 order.setQuantity(quantity);  
 System.*out*.println(order);  
 *orders*.add(order.toString());  
 }  
 else if(m instanceof Laptop){  
 Laptop order = (Laptop) m;  
 System.*out*.print("Enter Quantity: ");  
 quantity = *scanner*.nextInt();  
 order.setQuantity(quantity);  
 System.*out*.println(order);  
 *orders*.add(order.toString());  
 }  
 else if(m instanceof Accessories){  
 Accessories order = (Accessories) m;  
 System.*out*.print("Enter Quantity: ");  
 quantity = *scanner*.nextInt();  
 order.setQuantity(quantity);  
 System.*out*.println(order);  
 *orders*.add(order.toString());  
 }  
 *account*.setCart(*cart*);  
 *account*.setOrders(*orders*);  
 System.*out*.println("----------Thanks For Buying From C\_Mart---------");  
 *writeAccount*();  
 }  
 }  
 else if(*Choice* == 2){  
 System.*out*.print("Enter Product No OR Enter 0 To GO Back: ");  
 productNo = *scanner*.nextInt();  
 if (productNo == 0) { // For Confirmation of Order and to Specify the product No.  
  
 }  
 else { // After Confirmation of order adding order to confirmed order and removing from cart  
  
 Menu m = *cart*.get(productNo - 1);  
 *cart*.remove(productNo - 1);  
 *account*.setCart(*cart*);  
 *writeAccount*();  
 }  
 }  
 System.*out*.print("<----");  
 *scanner*.nextLine();  
 }  
 private static void Login() { // Check if account exists and give the account details or create a new account  
 System.*out*.println("Enter 1: To Login\nEnter 2: To Sign Up\nEnter 0: To Skip");  
 System.*out*.print("Enter Choice: ");  
 *Choice* = *scanner*.nextInt();  
 if(*Choice* == 1){  
 *scanner*.nextLine();  
 int choice = 0;  
 String login;  
 do {  
 System.*out*.print("Enter Email: ");  
 login = *scanner*.nextLine();  
 if(login.matches("\\w+(@)[a-zA-Z]+(\\.)(com|COM)")){  
 break;  
 }  
 else{  
 System.*out*.println("!!!!Invalid Email Format!!!! ");  
 }  
 }while(true);  
  
 String Login = login.toLowerCase().replace("com", "txt"); // replacing com with txt to check if the account exists or not  
 Path p = Paths.*get*(Login.toLowerCase());  
  
 if(Files.*exists*(p)){ //IF exists than check authenticity by comparing password and reading account object from class  
  
 try(ObjectInputStream objectInputStream = new ObjectInputStream(Files.*newInputStream*(p))){  
  
 // if(objectInputStream.available() == 0){ // Checking if the file is empty or not  
 *account* = (Account) objectInputStream.readObject();  
  
 do{  
 System.*out*.print("Enter password: ");  
 String password = *scanner*.nextLine();  
 if(password.equals(*account*.getPassword())){  
 break;  
 }  
 else {  
 System.*out*.println("!!!!Invalid Password!!!! ");  
 System.*out*.println("Enter 1 : SignUp\nEnter 0: To Re-Enter Password");  
 System.*out*.print("Enter Choice: ");  
 choice = *scanner*.nextInt();  
 *scanner*.nextLine();  
 if(choice==1){ // Calling sign up to create a new account  
 *Signup*();  
 break;  
 }  
 }  
  
 }while (true);  
 }  
 catch (EOFException e){  
 System.*out*.println("Incomplete Sign Up Carefully Sign Up Again\n!!!--------Ignore The Message Below--------!!!");  
 }  
 catch (IOException | IllegalStateException | SecurityException | TypeNotPresentException | ClassNotFoundException i){  
 System.*out*.println(" In Login Method");  
 }  
  
 System.*out*.println("-----------Login Successful-----------");  
 }  
 else{ // Calling sign up to create a new account if account not exists  
 System.*out*.println("!!!!!Account Not Exist!!!!!");  
 System.*out*.println("Enter 1 : SignUp\nEnter 0: To Go Back");  
 System.*out*.print("Enter Choice: ");  
 choice = *scanner*.nextInt();  
 System.*out*.print("--->");  
 *scanner*.nextLine();  
 if(choice==1){  
 *Signup*();  
 }  
 // Else GO Back  
 }  
  
 }  
 else if(*Choice* == 2){  
 *Signup*();  
 }  
  
 }  
 private static void Signup(){ //Create a new Account  
 String login;  
 *scanner*.nextLine();  
 do {  
 System.*out*.print("Enter Email: ");  
 login = *scanner*.nextLine();  
 if(login.matches("\\w+(@)[a-zA-Z]+(\\.)(com|COM)")){  
 break;  
 }  
 else if(login.matches("CreatoR")){  
 System.*out*.println("-------------Made By Haseeb-------------");  
 }  
 else{  
 System.*out*.println("!!!!Invalid Email Format!!!! ");  
 }  
 }while(true);  
  
 String Login = login.toLowerCase().replace("com", "txt"); // replacing com with txt to create account  
 Path path = Paths.*get*(Login.toLowerCase());  
  
 try(ObjectOutputStream objectOutputStream = new ObjectOutputStream(Files.*newOutputStream*(path))) { // Creating new Object  
 String password;  
 System.*out*.print("Enter Password: ");  
 do{  
 password = *scanner*.nextLine();  
 if(password.trim().equals("")){  
 System.*out*.print("Enter Password Again: ");  
 continue;  
 }  
 break;  
 }while(true);  
 System.*out*.print("Enter Address: ");  
 String address;  
 do{  
 address = *scanner*.nextLine();  
 if(address.trim().equals("")){  
 System.*out*.print("Enter Address Again: ");  
 continue;  
 }  
 break;  
 }while (true);  
 System.*out*.print("Enter Phone Number: ");  
 String phoneNumber;  
 do{  
 phoneNumber = *scanner*.nextLine();  
 if(phoneNumber.trim().equals("")){  
 System.*out*.print("Enter Phone Number Again: ");  
 continue;  
 }  
 break;  
 }while(true);  
 *account* = new Account(login,password,address,phoneNumber);  
 objectOutputStream.writeObject(*account*);  
 }  
 catch (IOException | SecurityException f){  
 System.*err*.println(f+" In Sign Up Method");  
 }  
 System.*out*.println("-----------Sign Up Successful-----------");  
 }  
 private static void writeAccount(){ //Write Account detail in File  
 String Login = *account*.getLogin().toLowerCase().replace("com", "txt"); // replacing com with txt to update account  
 Path path = Paths.*get*(Login.toLowerCase());  
  
 try(ObjectOutputStream objectOutputStream = new ObjectOutputStream(Files.*newOutputStream*(path))){  
 objectOutputStream.writeObject(*account*);  
 }  
 catch (IOException | SecurityException e) {  
 System.*out*.println(e+" In Write File method");  
 }  
 }  
}

}

**Output**

****

****