```
/*-------------------------------------------------------
My name:Alex Aiello
My student number:7223638
My course code: CSIT121
My email address:aaa163@uow.edu.au
Assignment number: 2
-------------------------------------------------------*/

import java.util.ArrayList;

import java.util.*;




class ComputerParts{

	private final String name;


	private final String brand;


	private final String socket;


	private final String size;


	private final String modelNo;


	private double price;


	public ComputerParts(String name , String brand, String socket, String modelNo, String size,
double price) {



		this.name = name;


		this.brand = brand;


		this.socket = socket;
```

```
                this.size = size;

                this.modelNo = modelNo;

                this.price = price;

        }
public String getName(){
        return name;

        }

        public String getBrand() {

                return brand;

        }

        public String getSocket() {

                return socket;

        }

        public String getModelNo() {

                return modelNo;

        }
```

```
	public String getSize() {

		return size;

	}

	public double getPrice() {

		return price;

	}

	public void setPrice(int price) {

		this.price = price;

	}

	public String toString() {
		return name +"\t" + socket + "\t" + brand + "\t" + "\t" + modelNo +  size +"\t" +
price;

	}



}

class Customer {
	private final String ID;
	private String name;
	private String gender;
	private String address;
```

```
private long mobileNumber;
public Customer(String ID, String name, String gender, String address,
                    long mobileNumber) {
        Random r = new Random(System.currentTimeMillis());
        this.ID = "C" + 0 + ((1 + r.nextInt(2)) * 10000 + r.nextInt(10000));
        this.name = name;
        this.gender = gender;
        this.address = address;
        this.mobileNumber = mobileNumber;
}
public String getID() {
        return ID;
}
public String getName() {
        return name;
}
public void setName(String name) {
        this.name = name;
}
public String getGender() {
        return gender;
}
public void setGender(String gender) {
        this.gender = gender;
}
public String getAddress() {
        return address;
}
public void setAddress(String address) {
        this.address = address;
}
```

```
	public long getMobileNumber() {
		return mobileNumber;
	}
	public void setMobileNumber(int mobileNumber) {
		this.mobileNumber = mobileNumber;
	}
	public String toString() {
return "Customer's Info:- " + "\n" +

"\tID : " + ID + "\n" +

"\tName : " + name + "\n" +

"\tGender : " + gender + "\n" +

"\tMobile No : " + mobileNumber + "\n" +

"\tAddress : " + address;
}

}

class Order {
	private ComputerParts computerparts;
	private String id;
   private double totalPrice;
	private final Customer customerInfo;
	public Order(Customer customerInfo) {
		Random r = new Random(System.currentTimeMillis());
		this.id = "O" + 0 + ((1 + r.nextInt(2)) * 10000 + r.nextInt(10000));
		this.totalPrice = 0.0 * 0.05/ 100;
```

```
		this.customerInfo = customerInfo;
	}
	public String getId() {
		return id;
	}
	public void setId(String id) {
		this.id = id;
	}
	public double getTotalPrice() {
		return totalPrice;
	}
	public void setTotalPrice(double totalPrice) {
		this.totalPrice = totalPrice;
	}
	public ComputerParts getComputerParts() {
		return computerparts;
	}
	public void setComputerParts(ComputerParts computerparts) {
		this.computerparts = computerparts;
	}
	public Customer getCustomerInfo() {
		return customerInfo;
	}
	 public String toString() {
		StringBuilder builder = new StringBuilder();
		builder.append("Order ID: ").append(id).append("\n");
		if (computerparts != null) {
			builder.append("Computer Parts   \t:
").append(computerparts.toString()).append("\n");
			totalPrice = computerparts.getPrice();
		}
```

```
 builder.append("Total Price \t: ").append(totalPrice);
 return builder.toString();
}
}

class OrderingSystem {
	private Order currentOrder;
	private Customer newCustomer;
	private Customer memberCustomer;
	private ArrayList<Order> orderList;
	private final Scanner subMenuScanner;
	private final ArrayList<Customer> customersList;
	private ArrayList<ComputerParts> computerPartsList;

	public OrderingSystem() {
		subMenuScanner = new Scanner(System.in);
		this.orderList = new ArrayList<>();
		this.customersList = new ArrayList<>();
		this.computerPartsList = new ArrayList<>();
		AllLists();
	}

	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);
		OrderingSystem ob = new OrderingSystem();
		System.out.println(" \n---------- Welcome to the Shop ----------\n");
		ob.currentOrder = ob.getOrderList();
		// Display the Main Menu
		Label:
		while (true) {
			switch (displayMenu()) {
```

```
                case 1:
                    ob.newCustomer();
                        break;
            case 2:
                    ob.memberCustomer();
                        break;
                case 3:
                        ob.addComputerParts();
                        break;
            case 4:
                        ob.showOrderList(true);
                        break;
                case 5:
                        ob.addToOrderList(ob);
                        break;
                case 6:
                    ob.editSubmittedOrder(true);
                        break;
                case 7:
                        System.out.println("Thankyou For Your Purchase.");
                        break Label;
                default:
                        System.out.println("Invalid input! Please try again.");
                        break;
                }
        }
}

private void addToOrderList(OrderingSystem ob) {
        if (currentOrder.getTotalPrice() > 0) {
                orderList.add(currentOrder);
```

```
                System.out.println("Your product list added into the Order list.");


                ob.showOrderList(false);
        }
 System.out.print("Thankyou your product has been added to your order");
}


private void addComputerParts() {
        Label:
        while (true) {
                int selectedComputerParts = displayComputerPartsTable();
                switch (selectedComputerParts) {
                case 0:
                  break Label;
                case 1:
                case 2:
                case 3:
                case 4:
                case 5:
                case 6:
                case 7:
                case 8:
                case 9:
                case 10:
                case 11:
                case 12:
                case 13:
                case 14:
                case 15:
                case 16:
                case 17:
```

```
                        case 18:
                        case 19:
                        case 20:
                        case 21:
                        case 22:
                        case 23:
                        case 24:
                        case 25:
                        case 26:
                        case 27:
                        case 28:
                        case 29:
                        case 30:
                        case 31:
                        case 32:
                        case 33:
                        case 34:
                        case 35:
                                ComputerParts computerparts =
computerPartsList.get(selectedComputerParts - 0);
                                currentOrder.setComputerParts(computerparts);
                                currentOrder.setTotalPrice(currentOrder.getTotalPrice() +
computerparts.getPrice());
                                System.out.printf("\n%s model Number %s successfully added to
cart.\n" , computerparts.getBrand(),computerparts.getModelNo());
        break Label;
                        default:
                                System.out.println("Invalid input! Please try again.");
                                break;
                        }
                }
        }
```

```
private int displayComputerPartsTable() {



            System.out.println("Available Computer Parts: ");
            System.out.println("-------------------------------------------");
            for (int i = 0; i < computerPartsList.size(); i++) {
            ComputerParts computerParts = computerPartsList.get(i);
            System.out.println((i + 1) + "\t" + computerParts.toString());
            }
      System.out.print("Select a Computer Part (0 for main menu) : ");
                  return subMenuScanner.nextInt();
}



private void showOrderList(boolean modifyOption) {
            System.out.println("\nYour Order List: ");
            System.out.println(currentOrder.toString());
            System.out.println();
            System.out.println(memberCustomer.toString());

            if(modifyOption){
                  modifyOrder();
            }
}

//this method to modify the order
private void modifyOrder() {
            subMenuScanner.nextLine();
            System.out.println("\nDo you want to modify your order? (Y/N)");
            String modify = subMenuScanner.nextLine();
            if("Y".equals(modify.toUpperCase())) {
```

```
                    System.out.print("Enter computer component ID to Be remove: ");



                    int componentToRemove = subMenuScanner.nextInt();
                    switch(componentToRemove) {
                    case 1:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

          currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 2:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

          currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 3:
                    if(currentOrder.getComputerParts() == null) {
```

```
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 4:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 5:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
```

```
                                        }
                                        break;
                                case 6:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 7:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 8:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {
```

```
        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 9:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 10:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 11:
```

```
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 12:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 13:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
```

```
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 14:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 15:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 16:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {
```

```
            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 17:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 18:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 19:
```

```
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 20:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 21:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
```

```
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 22:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 23:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 24:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {
```

```
        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 25:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 26:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 27:
```

```
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 28:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                    case 29:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
```

```
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 30:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 31:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current order list");
                                    }
                                    break;
                            case 32:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current order list");
                                    } else {
```

```
            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 33:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 34:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current order list");
                                        } else {

            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current order list");
                                        }
                                        break;
                                case 35:
```

```
                default:
                        System.out.println("Invalid selection");
                        break;
                }
        }
}

private void AllLists() {
        Random r = new Random(System.currentTimeMillis());
        computerPartsList.add(new ComputerParts("INTCPU"+0 + ((1 + r.nextInt(2)) * 10000
+ r.nextInt(10000)),"Intel","","i5","9600K",323));
        computerPartsList.add(new ComputerParts("INTCPU"+0 + ((1 + r.nextInt(2)) * 10000
+ r.nextInt(10000)),"Intel","","i7","9700K",462));
        computerPartsList.add(new ComputerParts("INTCPU"+0 + ((1 + r.nextInt(2)) * 10000
+ r.nextInt(10000)),"Intel","","i7","9700F",396));
        computerPartsList.add(new ComputerParts("INTCPU"+0 + ((1 + r.nextInt(2)) * 10000
+ r.nextInt(10000)),"Intel","","i9","9900K",591));
        computerPartsList.add(new ComputerParts("AMDCPU"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","4","Ryzen 2200",200));
        computerPartsList.add(new ComputerParts("AMDCPU"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","6","Ryzen 3600",310));
        computerPartsList.add(new ComputerParts("AMDCPU"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","8","Ryzen 3700",489));
        computerPartsList.add(new ComputerParts("AMDCPU"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","8","Ryzen 5800",669));
        computerPartsList.add(new ComputerParts("INTMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"Intel","","Gigabyte","H81M-DS2",129));
        computerPartsList.add(new ComputerParts("INTMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"Intel","","Asus","J40051-C",169));
        computerPartsList.add(new ComputerParts("INTMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"Intel","","MSI","Mpg-2390",225));
        computerPartsList.add(new ComputerParts("INTMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"Intel","","Gigabyte","Z490",471));
```

```
computerPartsList.add(new ComputerParts("AMDMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","Gigabyte","B-450",117));

computerPartsList.add(new ComputerParts("AMDMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","Asus","A320I",128));

computerPartsList.add(new ComputerParts("AMDMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","MSI","B450",232));

computerPartsList.add(new ComputerParts("AMDMOT"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","Gigabyte","X570S",679));

computerPartsList.add(new ComputerParts("Mon"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"Acer","","K242HYLB","24",194));

computerPartsList.add(new ComputerParts("Mon"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"LG","","32QN600","32",506));

computerPartsList.add(new ComputerParts("Mon"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"Asus","","MB16ACZ","16",429));

computerPartsList.add(new ComputerParts("Mon"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"MSI","","MP271QP","27",399));

computerPartsList.add(new ComputerParts("Mon"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"BenQ","","PD3200Q","32",653));

computerPartsList.add(new ComputerParts("Mon"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"Philips","","272M8CZ","27",289));

computerPartsList.add(new ComputerParts("INTGRA"+0 + ((1 + r.nextInt(2)) * 10000
+ r.nextInt(10000)),"NVIDIA","","Gigabyte","GeForce RTX 3070",1999));

computerPartsList.add(new ComputerParts("INTGRA"+0 + ((1 + r.nextInt(2)) * 10000
+ r.nextInt(10000)),"NVIDIA","","Asus","GeForceRTX3070",1899));

computerPartsList.add(new ComputerParts("INTGRA"+0 + ((1 + r.nextInt(2)) * 10000
+ r.nextInt(10000)),"NVIDIA","","MSI","GeForceRTX3080",3099));

computerPartsList.add(new ComputerParts("AMDGRA"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","Gigabyte","RadeonRX6900",3699));

computerPartsList.add(new ComputerParts("AMDGRA"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","Asus", "RadeonRX6900",3199));

computerPartsList.add(new ComputerParts("AMDGRA"+0 + ((1 + r.nextInt(2)) *
10000 + r.nextInt(10000)),"AMD","","MSI","RadeonRX6900",2699));

    computerPartsList.add(new ComputerParts("MEM"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"DDR3","Kingston","8G","KCP316ND8",116));

computerPartsList.add(new ComputerParts("MEM"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"DDR3","ADATA ","16G","AX4U360038G18",189));

computerPartsList.add(new ComputerParts("MEM"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"DDR3","G.Skill","8G ","F3-10666CL9D",96));
```

```
		computerPartsList.add(new ComputerParts("MEM"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"DDR4","Kingston","8G ","KCP426SS8",93));
		computerPartsList.add(new ComputerParts("MEM"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"DDR4","G.Skill ","16G","F4-266C18S",158));
		computerPartsList.add(new ComputerParts("MEM"+0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)),"DDR4","Crucial","32G","CT32G4SFD832",159));
	}

	private Order getOrderList() {
		for (Order order : orderList) {
			if (order.getCustomerInfo().getName().equals(memberCustomer.getID())) {
				return order;
			}
		}
		return null;
	}

	 private void editSubmittedOrder(boolean modifyOption) {
		System.out.println("\nYour Order List: ");
		System.out.println(currentOrder.toString());
		System.out.println();
		System.out.println(memberCustomer.toString());

	   subMenuScanner.nextLine();
		System.out.println("\nDo you want to modify your order? (Y/N)");
		String modify = subMenuScanner.nextLine();
		if("Y".equals(modify.toUpperCase())) {
			System.out.print("Enter computer component ID to Be remove: ");

			int componentToRemove = subMenuScanner.nextInt();
```

```
                    switch(componentToRemove) {
                    case 1:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 2:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 3:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
```

```
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 4:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 5:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 6:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
```

```
                                        } else {

            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                        case 7:
                        if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                        case 8:
                        if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

            currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
```

```
                    case 9:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 10:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 11:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
```

```
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 12:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 13:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 14:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {
```

```
        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                        case 15:
                        if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                        case 16:
                        if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                        case 17:
```

```
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 18:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 19:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
```

```
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 20:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 21:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 22:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {
```

```
        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                                case 23:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                                case 24:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                                case 25:
```

```
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 26:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                    case 27:
                    if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
```

```
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 28:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 29:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                    }
                                    break;
                            case 30:
                            if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                    } else {
```

```
        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                                case 31:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                                case 32:
                                if(currentOrder.getComputerParts() == null) {
                                                System.out.println("No Computer Parts in the
current submitted order list");
                                        } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                                currentOrder.setComputerParts(null);
                                                System.out.println("Computer Parts removed from
the current submitted order list");
                                        }
                                        break;
                                case 33:
```

```
                        if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                }
                                break;
                        case 34:
                        if(currentOrder.getComputerParts() == null) {
                                        System.out.println("No Computer Parts in the
current submitted order list");
                                } else {

        currentOrder.setTotalPrice(currentOrder.getTotalPrice() -
currentOrder.getComputerParts().getPrice());
                                        currentOrder.setComputerParts(null);
                                        System.out.println("Computer Parts removed from
the current submitted order list");
                                }
                                break;
                        case 35:
                                default:
                                        System.out.println("Invalid selection");
                                        break;
                        }
                }
        }
        private static int displayMenu() {
                System.out.println("\n1. New Customer Form");
```

```
                System.out.println("2. Member Customer Login");
                System.out.println("3. Computer Parts");
                System.out.println("4. View and Edit Order");
                System.out.println("5. Submit the Order");
                System.out.println("6. View and Edit Submitted Order");
                System.out.println("7. Exit without the purchase");
                System.out.print("\nEnter your choice (1-7): ");
                return new Scanner(System.in).nextInt();
        }


        private void memberCustomer(){
        System.out.println("Member Customer: ");
                System.out.println("-------------------------------------------");
                for (int i = 0; i < customersList.size(); i++) {
                System.out.println( memberCustomer.toString());
                }
                        System.out.print("Select a Customer (0 for main menu) : ");
                        subMenuScanner.nextInt();


        }


        private void newCustomer() {
      Scanner sc = new Scanner(System.in);
                 Random r = new Random(System.currentTimeMillis());
                System.out.println("Your ID IS:"+"\t"+ "C" + 0 + ((1 + r.nextInt(2)) * 10000 +
r.nextInt(10000)));
                String id = sc.nextLine();
                System.out.print("Enter your name: ");
                String name = sc.nextLine();
                System.out.print("Enter your gender: ");
                String gender = sc.nextLine();
```

```
                System.out.print("Enter your address: ");
                String address = sc.nextLine();
                System.out.print("Enter your mobile number: ");
                long number = sc.nextLong();
                System.out.println("-----------------------------------------------");
                System.out.println("You Have Been Added As A Member ");
            memberCustomer = new Customer(id, name, gender, address, number) ;
                customersList.add(memberCustomer);
      currentOrder = new Order(memberCustomer);
          }
}
```