

يحيى محمود زكريا عبدالله Name:

ID: 23011619

## Screen shots of 8 classes

### 1- Product class

```
1  public class Product {
2      // attributes
3      private int productID; // positive else take absolute
4      private String name;
5      private float price; // positive else take absolute
6
7      // methods
8
9      // constructor
10     public Product(int productID, String name, float price) {
11         this.productID = Math.abs(productID);
12         this.name = name;
13         this.price = Math.abs(price);
14     }
15
16     // getter & setter
17     public int getProductID() {
18         return productID;
19     }
20     public String getName() {
21         return name;
22     }
23     public float getPrice() {
24         return price;
25     }
26
27     public void setProductId(int newId) {
28         productID = Math.abs(newId);
29     }
30     public void setName(String newName) {
31         name = newName;
32     }
33     public void setPrice(float newPrice) {
34         price = Math.abs(newPrice);
35     }
36 }
37
```

## 2- ElectronicProduct class

```
1 public class ElectronicProduct extends Product {  
2     // attributes  
3     // productID, name, price from "Product" class  
4     private String brand;  
5     private int warrantyPeriod;// positive else take absolute  
6  
7     // methods  
8  
9     // constructor  
10    public ElectronicProduct(int productID, String name, float price, String brand, int warrantyPeriod) {  
11        super(productID, name, price);  
12        this.brand = brand;  
13        this.warrantyPeriod = Math.abs(warrantyPeriod);  
14    }  
15  
16    // getter & setter  
17    // + getter and setter from "Product" class  
18    public String getBrand() {  
19        return brand;  
20    }  
21    public int getWarrantyPeriod() {  
22        return warrantyPeriod;  
23    }  
24  
25    public void setBrand(String newBrand) {  
26        brand = newBrand;  
27    }  
28    public void setWarrantyPeriod(int newWarrantyPeriod) {  
29        warrantyPeriod = Math.abs(newWarrantyPeriod);  
30    }  
31}  
32}
```

### 3- ClothingProduct class

```
1  public class ClothingProduct extends Product {
2      // attributes
3      private String size, fabric;
4
5      // constructor
6      public ClothingProduct(int productID, String name, float price, String size, String fabric) {
7          super(productID, name, price);
8          this.size = size;
9          this.fabric = fabric;
10     }
11
12     // getters and setters
13     public String getSize() {
14         return size;
15     }
16     public String getFabric() {
17         return fabric;
18     }
19
20     public void setSize(String newSize) {
21         size = newSize;
22     }
23     public void setFabric(String newFabric) {
24         fabric = newFabric;
25     }
26 }
27
```

#### 4- BookProduct class

```
1  public class BookProduct extends Product {
2      // attributes
3      private String author, publisher;
4
5      // constructor
6      public BookProduct(int productID, String name, float price, String author, String publisher) {
7          super(productID, name, price);
8          this.author = author;
9          this.publisher = publisher;
10     }
11
12     // getters and setters
13     public String getAuthor() {
14         return author;
15     }
16     public String getPublisher() {
17         return publisher;
18     }
19
20     public void setAuthor(String newAuthor) {
21         author = newAuthor;
22     }
23     public void setPublisher(String newPublisher) {
24         publisher = newPublisher;
25     }
26 }
27
```

## 5- Customer class

```
Customer.java -> Customer -> setAddress(String)
1  public class Customer {
2      // attributes
3      private int customerID; // positive else take absolute
4      private String name;
5      private String address;
6
7      // methods
8
9      // constructor
10     public Customer(int customerID, String name, String address) {
11         this.customerID = Math.abs(customerID);
12         this.name = name;
13         this.address = address;
14     }
15
16     // getter & setter
17     public int getCustomerId() {
18         return customerID;
19     }
20     public String getName() {
21         return name;
22     }
23     public String getAddress() {
24         return address;
25     }
26
27     public void setCustomerId(int newId) {
28         customerID = Math.abs(newId);
29     }
30     public void setName(String newName) {
31         name = newName;
32     }
33     public void setAddress(String newAddress) {
34         address = newAddress;
35     }
36 }
37
```

## 6- Cart class

```
public class Cart {  
    // attributes  
    private int customerId, nProducts;  
    private Product[] products;  
  
    // constructor  
    public Cart(int customerId, int nProducts) {  
        this.customerId = Math.abs(customerId); // to take absolute value  
        this.nProducts = Math.abs(nProducts); // to take absolute value  
        products = new Product[nProducts];  
    }  
  
    // getters and setters  
    public int getCustomerId() {  
        return customerId;  
    }  
    public int getNProducts() {  
        return nProducts;  
    }  
    public Product[] getProducts() {  
        return products;  
    }  
  
    public void setCustomerId(int newCustomerId) {  
        customerId = Math.abs(newCustomerId);  
    }  
    public void setNProducts(int newNProducts) {  
        nProducts = Math.abs(newNProducts);  
    }  
    public void setProducts(Product[] newProducts) {  
        products = newProducts;  
    }  
  
    public void addProduct(Product p) {
```

```
// add new product to the first null value
for(int i = 0; i < products.length; i++) {

    if(products[i] == null) {
        products[i] = p;

        return; // to stop the function after adding the product
    }

}

public void removeProduct(int index) {

    // to check if index is in the right range
    if(index < nProducts && index >= 0) {
        products[index] = null;
    }
}

public float calculatePrice() {
    float sum = 0;

    for(Product p : products) {
        if(p != null) {
            sum += p.getPrice();
        }
    }

    return sum;
}

public void placeOrder() {
    // place this order
    new Order(customerId, calculatePrice(), products);
}
```

## 7- Order class

```
import java.awt.GridLayout;

import javax.swing.BoxLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

public class Order {

    // attributes
    private int customerId;
    private float totalPrice;
    private Product[] products;

    // order id is starting from 1 and increases by one when you place
    // an order to be unique
    private static int orderId = 1;

    // for GUI
    JFrame detailsFrame;

    // methods

    // constructor
    public Order(int customerId, float totalPrice, Product[] products) {
        this.customerId = Math.abs(customerId);
        this.totalPrice = Math.abs(totalPrice);
        this.products = products;

        // open new window with order details
        detailsFrame = new JFrame("Order details");

        // show details once you click place order button
        printOrderInfo();

        detailsFrame.setVisible(true);
    }
}
```

```
    detailsFrame.pack();
}

// loads all details of order to GUI
public void printOrderInfo() {
    JPanel detailsPanel = new JPanel();
    JPanel productsPanel = new JPanel();
    productsPanel.setLayout(new GridLayout(products.length, 1, 10,
10));
    productsPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
    detailsPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
    detailsPanel.setLayout(new BoxLayout(detailsPanel,
BoxLayout.Y_AXIS));

    JLabel title = new JLabel("Here is your order's summary : ");

    // order id is starting from 1 and increases by one when you place
    an order to be unique
    JLabel id = new JLabel("Order ID : " + (orderId++));

    JLabel customer = new JLabel("Customer ID : " + customerId);

    detailsPanel.add(title);
    detailsPanel.add(id);
    detailsPanel.add(customer);

    detailsPanel.add(new JLabel("Products : "));

    for(int i = 0; i < products.length; i++) {

        if(products[i] != null) {
            JPanel p = new JPanel();
            p.setLayout(new GridLayout(1, 2));

            p.add(new JLabel(products[i].getName()));
            p.add(new JLabel("$ " + products[i].getPrice()));

            productsPanel.add(p);
        }
    }
}
```

```
        }

        detailsPanel.add(productsPanel);

        JLabel totalPriceLabel = new JLabel("Total Price : $ " +
totalPrice);
        detailsPanel.add(totalPriceLabel);

        detailsFrame.add(detailsPanel);
    }
}
```

## 8- EcommerceSystem class

```
// This file contains most of GUI and objects

import javax.swing.*;
import javax.swing.border.EmptyBorder;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

public class EcommerceSystem implements ActionListener {
    // content for GUI (to make it global as we can access it in any
function in the class)

    // Frames
    JFrame starterFrame, contentFrame;

    // Panels
    JPanel createUser, electronic, book, clothing, cartPanel,
cartProductsPanel;

    // TextFields
    JTextField NameField, idField, addressField, nProduct;

    //buttons
    JButton createBtn;

    // titles
    JLabel title;

    // to store all customers data
    ArrayList<Customer> customers = new ArrayList<>();
    static Cart cart;

    // products arrays
    ElectronicProduct[] eProducts = {
        new ElectronicProduct(0, "Laptop", 20000, "Asus", 2),
```

```

new ElectronicProduct(1, "Headphone", 500, "", 1),
new ElectronicProduct(2, "mouse", 100, "hp", 1),
new ElectronicProduct(3, "I phone 12", 30000, "I phone", 2),
};

BookProduct[] bProducts = {
    new BookProduct(4, "ابن كثير", "البداية والنهاية", 2500, ("دار النشر 1")),
    new BookProduct(5, "ابن القيم", "زاد المعاد في هدى خير العباد", 1500, ("دار النشر 1")),
    new BookProduct(6, "ابن الجوزي", "صيد الخاطر", 200, ("دار النشر 1")),
    new BookProduct(7, "أبو بكر الجزائري", "منهج المسلم", 200, ("دار النشر 1")),
};

ClothingProduct[] cProducts = {
    new ClothingProduct(8, "السبايدر", "قطن", 100.001f, "M"),
    new ClothingProduct(9, "T_Shirt polo", "قطن", 400.99f, "L"),
    new ClothingProduct(10, "Jeans", "600.5f", "34", "jeans"),
    new ClothingProduct(11, "socks", "99.99f", "44", "cotton"),
};

// constructor (initializes GUI)
public EcommerceSystem() {
    // create customer window GUI
    createUser = new JPanel();
    createUser.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
    createUser.setPreferredSize(new Dimension(300, 400));
    createUser.setBackground(Color.LIGHT_GRAY);

    // title
    title = new JLabel("Welcome to E-commerce System !");
    title.setFont(new Font("Arial", Font.PLAIN, 16));

    // id textflied and its label
    JPanel idPanel = new JPanel();
    idPanel.setBackground(Color.LIGHT_GRAY);
    idPanel.setPreferredSize(new Dimension(300, 50));
    JLabel idLabel = new JLabel("ID : ");
    idField = new JTextField("", 17);
    idPanel.add(idLabel);
    idPanel.add(idField);

    // name textfield and its label

```

```
JPanel namePanel = new JPanel();
namePanel.setBackground(Color.LIGHT_GRAY);
namePanel.setPreferredSize(new Dimension(300, 50));
JLabel nameLabel = new JLabel("Name :   ");
NameField = new JTextField("",17);
namePanel.add(nameLabel);
namePanel.add(NameField);

// address textfield and its label
JPanel addressPanel = new JPanel();
addressPanel.setBackground(Color.LIGHT_GRAY);
addressPanel.setPreferredSize(new Dimension(300, 50));
JLabel addressLabel = new JLabel("Address : ");
addressField = new JTextField("",17);
addressPanel.add(addressLabel);
addressPanel.add(addressField);

// number of products textfield and its label
JPanel nPanel = new JPanel();
nPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
nPanel.setBackground(Color.LIGHT_GRAY);
nPanel.setPreferredSize(new Dimension(300, 50));
JLabel nLabel = new JLabel("How many products you want to add to
your cart ?");
nProduct = new JTextField("",17);
nPanel.add(nLabel);
nPanel.add(nProduct);

// create customer button
createBtn = new JButton("Create");
createBtn.addActionListener(this);
createBtn.setBackground(Color.DARK_GRAY);
createBtn.setForeground(Color.white);

// add components to the panel and frame
createUser.add(title);
createUser.add(idPanel);
createUser.add(namePanel);
createUser.add(addressPanel);
```

```
createUser.add(nPanel);
createUser.add(createBtn);

starterFrame = new JFrame();
starterFrame.add(createUser, BorderLayout.CENTER);

starterFrame.pack();
starterFrame.setResizable(false);
starterFrame.setVisible(true);

///////////////////////////////
///

// products window
contentFrame = new JFrame("Products");
contentFrame.setLayout(new GridLayout(1, 4, 15, 0));

// electronic products panel
electronic = new JPanel();
electronic.setLayout(new GridLayout(8, 1, 10, 10));
JLabel eTitle = new JLabel("Electronic products");
eTitle.setFont(new Font("Arial", Font.BOLD, 16));
electronic.add(eTitle);

// book products panel
book = new JPanel();
book = new JPanel(new GridLayout(8, 1, 10, 10));
JLabel bTitle = new JLabel("Book products");
bTitle.setFont(new Font("Arial", Font.BOLD, 16));
book.add(bTitle);

// clothing products panel
clothing = new JPanel();
clothing = new JPanel(new GridLayout(8, 1, 10, 10));
JLabel cTitle = new JLabel("Clothing products");
cTitle.setFont(new Font("Arial", Font.BOLD, 16));
clothing.add(cTitle);

// shopping cart panel
```

```
cartPanel = new JPanel();
cartPanel.setLayout(new BoxLayout(cartPanel, BoxLayout.Y_AXIS));

// adding panels to the frame
contentFrame.add(electronic);
contentFrame.add(book);
contentFrame.add(clothing);
contentFrame.add(cartPanel);
contentFrame.setExtendedState(JFrame.MAXIMIZED_BOTH); // full
screen
contentFrame.setVisible(false); // to make it hidden untill we
create new customer

// add proutdcs UI (electronic)
for(ElectronicProduct p : eProducts) {
    JLabel labelName = new JLabel(p.getName());
    JLabel labelPrice = new JLabel(p.getPrice() + " $");
    JLabel labelId = new JLabel("#" + p.getProductId());

    JPanel product = new JPanel();
    product.setLayout(new GridLayout(2, 2));
    product.setBackground(Color.lightGray);
    product.setBorder(new EmptyBorder(0, 5, 0, 0));

    // what will happen when user clicks Add button
    JButton addToOrder = new JButton(new AbstractAction("Add") {
        @Override
        public void actionPerformed( ActionEvent e ) {
            cart.addProduct(p);
            updateCart(); // to update the cart UI
        }
    });

    addToOrder.setBackground(Color.DARK_GRAY);
    addToOrder.setForeground(Color.white);

    // add content to product panel and then add this panel to main
    product.add(labelId);
```

```
        product.add(labelName);
        product.add(labelPrice);
        product.add(addToOrder);
        electronic.add(product);
    }

    // add products UI (books)
    for(BookProduct p : bProducts) {
        JLabel labelName = new JLabel(p.getName());
        JLabel labelPrice = new JLabel(p.getPrice() + " $");
        JLabel labelId = new JLabel("#" + p.getProductId());

        JPanel product = new JPanel();
        product.setLayout(new GridLayout(2, 2));
        product.setBackground(Color.lightGray);
        product.setBorder(new EmptyBorder(0, 5, 0, 0));

        // what will happen when user clicks Add button
        JButton addToOrder = new JButton(new AbstractAction("Add") {
            @Override
            public void actionPerformed( ActionEvent e ) {
                cart.addProduct(p);
                updateCart(); // to update the cart UI
            }
        });

        addToOrder.setBackground(Color.DARK_GRAY);
        addToOrder.setForeground(Color.white);

        // add content to product panel and then add this panel to main
        product.add(labelId);
        product.add(labelName);
        product.add(labelPrice);
        product.add(addToOrder);
        book.add(product);
    }

    // add products UI (clothing)
```

```
for(ClothingProduct p : cProducts) {
    JLabel labelName = new JLabel(p.getName());
    JLabel labelPrice = new JLabel(p.getPrice() + " $");
    JLabel labelId = new JLabel("#" + p.getProductID());

    JPanel product = new JPanel();
    product.setLayout(new GridLayout(2, 2));
    product.setBackground(Color.lightGray);
    product.setBorder(new EmptyBorder(0, 5, 0, 0));

    // what will happen when user clicks Add button
    JButton addToOrder = new JButton(new AbstractAction("Add") {
        @Override
        public void actionPerformed( ActionEvent e ) {
            cart.addProduct(p);
            updateCart(); // to update the cart UI
        }
    });

    addToOrder.setBackground(Color.DARK_GRAY);
    addToOrder.setForeground(Color.white);

    // add content to product panel and then add this panel to main
    product.add(labelId);
    product.add(labelName);
    product.add(labelPrice);
    product.add(addToOrder);
    clothing.add(product);
}

public static void main(String[] args) {

    // all application starts from this point
    new EcommerceSystem();

}

// all button actions is here
```

```
@Override
public void actionPerformed(ActionEvent e) {

    if(e.getSource() == createBtn) {
        // create new customer, open products window and close create
        user window
        Customer newC = new
Customer(Integer.parseInt(idField.getText()), NameField.getText(),
addressField.getText());

        // add this customer to customers ArrayList
        customers.add(newC);

        // create new cart for this customer with specified number of
        products
        cart = new Cart(newC.getCustomerId(),
Integer.parseInt(nProduct.getText()));

        // to render cart content
        updateCart();

        // opens products page
        contentFrame.setVisible(true);

        // closes create customer page
        starterFrame.dispose();
    }
}

// renders cart UI
public void updateCart() {
    // to remove old UI
    cartPanel.removeAll();

    // adding all new UI to cart component
    JLabel cartTitle = new JLabel("Shopping Cart");
    cartTitle.setFont(new Font("Arial", Font.BOLD, 16));
    cartPanel.add(cartTitle);
```

```
JLabel cartSize = new JLabel("Your cart size is " +  
cart.getNProducts());  
cartSize.setFont(new Font("Arial", Font.BOLD, 17));  
cartPanel.add(cartSize);  
  
cartProductsPanel = new JPanel();  
// to make it as grid layout with number of rows = number of  
products chosen by customer  
cartProductsPanel.setLayout(new GridLayout(cart.getNProducts(), 1,  
10, 10));  
  
Product[] products = cart.getProducts();  
  
for(int i = 0; i < products.length; i++) {  
  
// to check if product is not removed from cart  
if(products[i] != null) {  
  
final int index = i; // as to not generate an error  
  
JPanel product = new JPanel();  
product.setLayout(new GridLayout(2, 2));  
product.setBackground(Color.lightGray);  
product.setBorder(new EmptyBorder(0, 5, 0, 0));  
  
JLabel pName = new JLabel(products[i].getName());  
JLabel pPrice = new JLabel(products[i].getPrice() + " $");  
JLabel pId = new JLabel("#" + products[i].getProductId());  
  
JButton delete = new JButton(new AbstractAction("remove") {  
@Override  
public void actionPerformed( ActionEvent e ) {  
  
// to remove product by its index from the cart  
cart.removeProduct(index);  
  
// update the new UI  
updateCart();  
}  
}
```

```
});

        delete.setBackground(Color.red);
        delete.setForeground(Color.white);

        product.add(pId);
        product.add(pName);
        product.add(pPrice);
        product.add(delete);
        cartProductsPanel.add(product);
    }
}

cartPanel.add(cartProductsPanel);

// add total price GUI
JLabel totalPriceLabel = new JLabel("Total price is " +
cart.calculatePrice() + " $");
totalPriceLabel.setFont(new Font("Arial", Font.BOLD, 17));
cartPanel.add(totalPriceLabel);

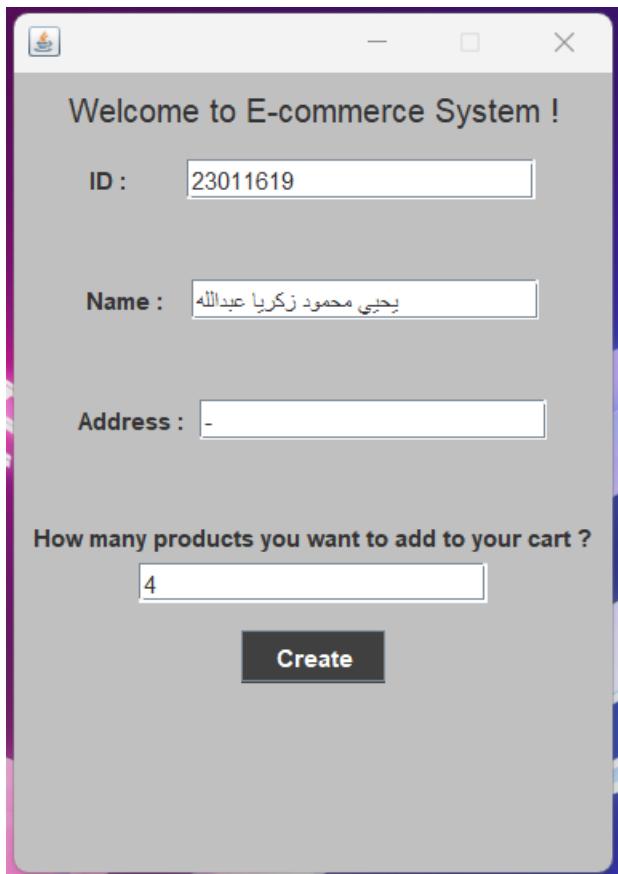
// place order button
JButton placeOrder = new JButton(new AbstractAction("placeOrder")
{
    @Override
    public void actionPerformed( ActionEvent e ) {

        // will create object from class Order and open new window
which contains all details
        cart.placeOrder();
    }
});
placeOrder.setBackground(Color.GREEN);
cartPanel.add(placeOrder);

// required to refresh the content
cartPanel.revalidate();
cartPanel.repaint();
}
}
```

## Screen shots of test case

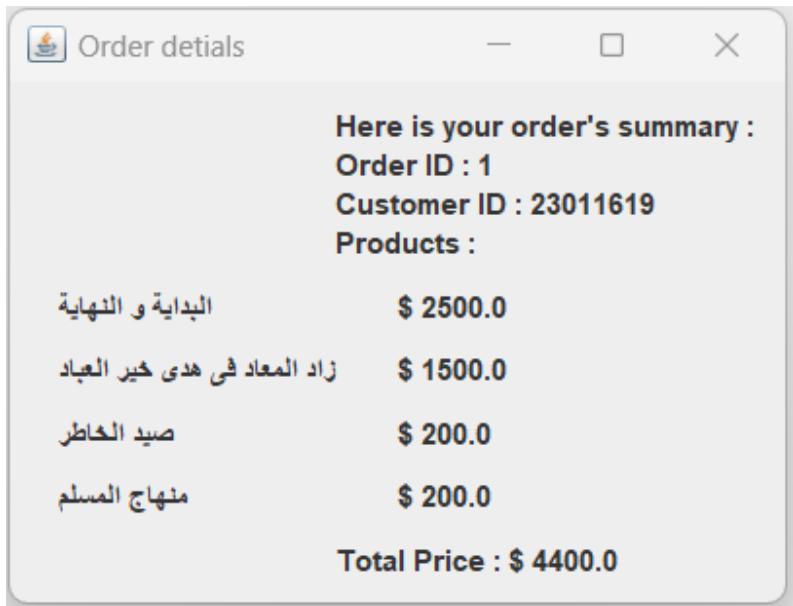
Create new customer:



After creating a customer and adding some products to the cart:

The screenshot shows a Windows desktop environment. On the left, there is a window titled "Products" containing three grids: "Electronic products", "Book products", and "Clothing products". The "Electronic products" grid has items #0 (Laptop, \$20000.0), #1 (Headphone, \$500.0), #2 (mouse, \$100.0), and #3 (iPhone 12, \$30000.0). The "Book products" grid has items #4 (البداية و النهاية, \$2500.0), #5 (زاد المعاد في هدى خير العباد, \$1500.0), #6 (صيد الحاضر, \$200.0), and #7 (منهج المسلم, \$200.0). The "Clothing products" grid has items #8 (السياندر, \$100.01), #9 (T\_Shirt polo, \$400.99), #10 (Jeans, \$600.5), and #11 (socks, \$99.99). Each item row contains an "Add" button. To the right, a "Shopping Cart" window is open, showing four items: #4 (\$2500.0, البداية و النهاية), #5 (\$1500.0, زاد المعاد في هدى خير العباد), #6 (\$200.0, صيد الحاضر), and #7 (\$200.0, منهاج المسلم). Each item in the cart has a "remove" button. At the bottom of the cart window, it says "Total price is 4400.0 \$" and has a green "placeOrder" button. The taskbar at the bottom shows various pinned icons and system status.

After clicking on place order button:



Thanks 😊 ...