

Case: Online Shopping Platform

Let's design an Online Shopping Platform with several classes, including an abstract class and interfaces, to manage users, products, and orders.

Class 1: User

- Attributes:
 - `userId (String)`: Unique identifier for each user.
 - `username (String)`: The username of the user.
 - `email (String)`: The email address of the user.
- Methods:
 - `displayUserInfo(): void`: Displays detailed information about the user, including user ID, username, and email.
 - `updateEmail(String newEmail): void`: Allows the user to update their email address.
 - `placeOrder(Order order): void`: Places an order on behalf of the user.

Class 2: Customer (extends User)

- Additional Attributes:
 - `address (String)`: The shipping address of the customer.
 - `orders (ArrayList<Order>)`: List of orders placed by the customer.
 - `cart (ArrayList<Product>)`: Shopping cart containing products selected by the customer.
- Additional Methods:
 - `displayCustomerInfo(): void`: Displays detailed information about the customer, including their address and order history.
 - `addToCart(Product product): void`: Adds a product to the customer's shopping cart.
 - `viewCart(): void`: Displays the contents of the customer's shopping cart.

Abstract Class: Product

- Attributes:
 - `productId (String)`: Unique identifier for each product.
 - `productName (String)`: The name of the product.
 - `price (double)`: The price of the product.
- Methods:
 - `displayProductInfo(): void`: Abstract method to display detailed information about the product.
 - `getPrice(): double`: Abstract method to get the price of the product.

- `isAvailable(): boolean`: Abstract method to check the availability of the product.

Interface: Orderable

- Methods:
 - `addToOrder(Order order): void`: Allows adding a product to an order.
 - `displayOrderDetails(): void`: Displays details of the order.

Class 3: Clothing (extends Product implements Orderable)

- Additional Attributes:
 - `size (String)`: The size of the clothing item.
 - `quantityInStock (int)`: The quantity of the clothing item available in stock.
- Additional Methods:
 - `displayProductInfo(): void`: Overrides the method in the base class to include information about the size.
 - `getPrice(): double`: Overrides the method in the base class to get the price of the clothing item.
 - `isAvailable(): boolean`: Overrides the method in the base class to check the availability of the clothing item.
 - `addToOrder(Order order): void`: Implements adding a clothing item to an order.
 - `displayOrderDetails(): void`: Implements displaying details of the order for a clothing item.

Class 4: Electronics (extends Product implements Orderable)

- Additional Attributes:
 - `brand (String)`: The brand of the electronic item.
 - `warrantyPeriod (int)`: The warranty period of the electronic item.
- Additional Methods:
 - `displayProductInfo(): void`: Overrides the method in the base class to include information about the brand.
 - `getPrice(): double`: Overrides the method in the base class to get the price of the electronic item.
 - `isAvailable(): boolean`: Overrides the method in the base class to check the availability of the electronic item.
 - `addToOrder(Order order): void`: Implements adding an electronic item to an order.
 - `displayOrderDetails(): void`: Implements displaying details of the order for an electronic item.

Class 5: Order

- **Attributes:**
 - `orderId (String)`: Unique identifier for each order.
 - `orderDate (String)`: The date when the order was placed.
 - `orderItems (ArrayList<Product>)`: List of products in the order.
 - `Status (String)`: The status of order. (Ordered, Shipped, Delivered)
- **Methods:**
 - `displayOrderInfo(): void`: Displays detailed information about the order, including order ID, order date, and order items.
 - `calculateTotalPrice(): double`: Calculates the total price of the order.
 - `processOrder(): void`: Processes the order, updating product quantities and notifying the customer.

Class 6: ShoppingSystem

- **Attributes:**
 - `products (ArrayList<Product>)`: List of available products in the shopping system.
 - `users (ArrayList<User>)`: List of registered users.
- **Methods:**
 - `displayMainMenu(): void`: Displays the main menu of the online shopping platform.
 - `displayProductsMenu(): void`: Displays options related to shopping for products.
 - `displayCartMenu(): void`: Displays options related to the customer's shopping cart.
 - `displayOrdersMenu(): void`: Displays options related to viewing orders.
 - `displayCustomerInfo(): void`: Displays information about the customer.
 - `exit(): void`: Exits the online shopping platform.

Screen Displays:

Input Customer ID:

```
=== Online Shopping Platform ===
Input Customer ID:
```

Main Menu:

```
=== Online Shopping Platform ===
1. Shop for Products
2. View Shopping Cart
3. View Orders
```

```
4. Customer Info
5. Exit
Your Choice:
```

Shop for Products Menu:

```
=== Shop for Products ===
1. View All Products
2. Add Clothing to Cart
3. Add Electronics to Cart
4. Back to Main Menu
Your Choice:
```

1. View All Products:

- Display a list of all available products with details such as product ID, name, price, and availability.
- Users can browse through the product list.

```
=== View All Products ===
ID    | Product Name          | Price   | Availability
-----
P001  | T-Shirt - Blue        | $19.99  | In Stock
P002  | Jeans - Slim Fit      | $39.99  | In Stock
P003  | Smartphone - Model X  | $499.99 | In Stock
P004  | Laptop - UltraBook    | $899.99 | In Stock
P005  | Sneakers - Sports     | $59.99  | In Stock
-----
Press Enter to go back to Shop for Products Menu...
```

2. Add Clothing to Cart:

- Display a list of available clothing items with details such as product ID, name, price, size, and availability.
- Prompt the user to enter the product ID of the clothing item they want to add to their shopping cart.
- Add the selected clothing item to the user's shopping cart.

```
=== Add Clothing to Cart ===
ID    | Clothing Name          | Price   | Size   | Availability
-----
P001  | T-Shirt - Blue        | $19.99  | M      | In Stock
```

```
P002 | Jeans - Slim Fit      | $39.99 | 32/34 | In Stock
P005 | Sneakers - Sports     | $59.99 | 9      | In Stock
```

Enter the ID to add to your cart (or type 'back' to go back):

3. Add Electronics to Cart:

- Display a list of available electronic items with details such as product ID, name, price, brand, and availability.
- Prompt the user to enter the product ID of the electronic item they want to add to their shopping cart.
- Add the selected electronic item to the user's shopping cart.

```
=== Add Electronics to Cart ===
```

```
ID    | Electronics Name          | Price   | Brand        | Availability
-----
P003  | Smartphone - Model X     | $499.99 | TechCo       | In Stock
P004  | Laptop - UltraBook       | $899.99 | MegaElect    | In Stock
P007  | Smartwatch - Fitness     | $129.99 | FitTech      | In Stock
```

Enter the ID to add to your cart (or type 'back' to go back):

4. Back to Main Menu:

- Return to the main menu.

View Shopping Cart Menu:

```
=== View Shopping Cart ===
```

```
1. View Cart Contents
2. Checkout
3. Back to Main Menu
```

Your Choice:

1. View Cart Contents:

- Display the contents of the user's shopping cart, including product details and quantity.
- Show the total price of the items in the cart.

```
=== View Cart Contents ===
```

```
ID    | Product Name              | Price
```

```
P001 | T-Shirt - Blue          | $19.99
P001 | T-Shirt - Blue          | $19.99
P003 | Smartphone - Model X    | $499.99
-----

Total Price: $539.97
Press Enter to go back to View Shopping Cart Menu...
```

2. Checkout:

- Process the checkout, which involves:
 - Displaying a summary of the order with product details and total price.
 - Prompting the user to confirm the order.
 - If confirmed, create a new order and update product quantities.
 - Notify the user that the order has been placed.

```
=== Checkout ===
Review your order:
ID    | Product Name          | Price
-----
P001  | T-Shirt - Blue        | $19.99
P001  | T-Shirt - Blue        | $19.99
P003  | Smartphone - Model X   | $499.99
-----

Total Price: $539.97
Do you want to proceed with the checkout? (Y/N):
```

3. Back to Main Menu:

- Return to the main menu.

View Orders Menu:

```
=== View Orders ===
1. View Order History
2. Track Order
3. Back to Main Menu
Your Choice:
```

1. View Order History:

- Display a list of the user's previous orders with details such as order ID, order date, and total price.
- Allow the user to select a specific order to view detailed order items.

```
=== View Order History ===
```

Order ID	Order Date	Total Price
O001	2023-12-01	\$919.98
O002	2023-12-05	\$799.98

```
Enter the Order ID to view details (or type 'back' to go back):
```

```
=== Order Details ===
```

```
Order ID: O001
```

```
Order Date: 2023-12-01
```

```
Total Price: $919.98
```

```
Order Items:
```

ID	Product Name	Price
P001	T-Shirt - Blue	\$19.99
P004	Laptop - UltraBook	\$899.99

```
Press Enter to go back to View Order History...
```

2. Track Order:

- Prompt the user to enter the order ID they want to track.
- Display the current status of the order.

```
=== Track Order ===
```

```
Enter the Order ID to track (or type 'back' to go back):
```

```
=== Order Tracking ===
```

```
Order ID: O001
```

```
Order Status: Shipped
```

```
Press Enter to go back to Track Order Menu...
```

3. Back to Main Menu:

- Return to the main menu.

Customer Info:

```
=== Customer Info ===  
Customer ID: C003  
Username    : mike_jones  
Email       : mike.jones@email.com  
Address     : 789 Pine Road, Villagetown  
Press Enter to back to Main Menu...
```

This Online Shopping Platform provides users with the ability to shop for products, manage their shopping cart, view order history, and check their customer information. Each menu option leads to specific functionalities within the system.