**Online Shopping System Project**

**Objective:**

The goal of this project is to create a simplified **Online Shopping System** using **Object-Oriented Programming** (OOP) concepts like classes, objects, constructors, methods, and arrays. You will implement classes for **Product**, **ShoppingCart**, **User**, and **Store** to simulate an online shopping experience.

**Project Breakdown:**

**1. Product Class**

The Product class will represent a product available for sale in the store.

**Attributes**:

* productName (String): The name of the product.
* productId (String): A unique ID for the product.
* price (double): The price of the product.
* quantityAvailable (int): The number of items available in the store.

**Methods**:

* getDetails(): Returns a string with the product's details (name, ID, price, and quantity).
* updateQuantity(int quantity): Decreases the product’s quantity when a purchase is made.

**2. ShoppingCart Class**

The ShoppingCart class represents a user's shopping cart, where products can be added and removed.

**Attributes**:

* cartItems[] (array of Product objects): An array that holds the products added to the cart.

**Methods**:

* addToCart(Product product): Adds a product to the shopping cart.
* removeFromCart(String productId): Removes a product from the cart by its ID.
* viewCart(): Displays all products in the cart, along with the total price.
* checkout(): Displays the total price and finalizes the checkout process.

**3. User Class**

The User class represents an individual user of the store.

**Attributes**:

* username (String): The username of the user.
* shoppingCart (ShoppingCart): An object that holds the user's shopping cart.

**Methods**:

* addItemToCart(Product product): Adds a product to the user’s shopping cart.
* removeItemFromCart(String productId): Removes a product from the user’s cart.
* viewShoppingCart(): Views the products in the user’s shopping cart.
* checkout(): Completes the checkout process.

**4. Store Class (Main Program)**

The Store class is responsible for managing products and users. It allows products to be viewed, added, and removed from the store's inventory.

**Attributes**:

* products[] (array of Product objects): An array that stores all the products available for sale in the store.

**Methods**:

* addProduct(Product product): Adds a product to the store’s inventory.
* removeProduct(String productId): Removes a product from the store.
* viewProducts(): Displays all available products.
* createUser(String username): Creates a new user.
* login(String username): Logs in an existing user based on their username.

**Instructions:**

**Step 1: Create the Product Class**

1. Define the Product class with the following attributes: productName, productId, price, and quantityAvailable.
2. Create a constructor that initializes the attributes.
3. Write a method called getDetails() that returns a string containing the product’s name, ID, price, and available quantity.
4. Write a method called updateQuantity(int quantity) that reduces the product’s available quantity when a product is purchased.

**Step 2: Create the ShoppingCart Class**

1. Define the ShoppingCart class. Create an array cartItems[] to hold up to 10 products.
2. Write a method called addToCart(Product product) to add a product to the shopping cart.
3. Write a method called removeFromCart(String productId) to remove a product from the shopping cart by its product ID.
4. Write a method called viewCart() to display the products in the shopping cart and calculate the total price.
5. Write a method called checkout() that displays the total price and confirms the checkout.

**Step 3: Create the User Class**

1. Define the User class with the attributes username and shoppingCart.
2. Write methods for adding items to the cart, removing items, and viewing the cart. Use the methods from the ShoppingCart class.
3. Write a checkout() method that prints the total price and simulates the completion of the purchase.

**Step 4: Create the Store Class**

1. Define the Store class with an array products[] to store products available in the store.
2. Write a method called addProduct(Product product) to add products to the store.
3. Write a method called removeProduct(String productId) to remove products by their ID.
4. Write a method called viewProducts() that displays all the available products in the store.
5. Write a method called createUser(String username) to create a new user.
6. Write a method called login(String username) to log in an existing user and assign them to the store.

**Step 5: Implement the Main Logic**

1. In the main() method, create a few sample products and add them to the store.
2. Create a user, add products to their shopping cart, view the cart, and simulate checkout.