Assignment 5

Let's walk through implementing this online shopping cart system. We'll define each class step-by-step according to the requirements.

Step 1: Implement the Product Class

```java
public class Product {
    private String productId;
    private String name;
    private double price;
    private int stockQuantity;

    public Product(String productId, String name, double price, int stockQuantity) {
        this.productId = productId;
        this.name = name;
        this.price = price;
        this.stockQuantity = stockQuantity;
    }

    public String getProductId() {
        return productId;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public int getStockQuantity() {
        return stockQuantity;
    }

    public void updateStockQuantity(int quantity) {
        this.stockQuantity += quantity;
    }

    @Override
    public String toString() {
        return "Product{" +
                "productId='" + productId + '\'' +
                ", name='" + name + '\'' +
                ", price=" + price +
                ", stockQuantity=" + stockQuantity +
                '}';
    }
}
```

Step 2: Implement the Customer Class

```java
import java.util.ArrayList;
import java.util.List;

public class Customer {
    private String customerId;
    private String name;
    private String email;
    private List<Product> cart;

    public Customer(String customerId, String name, String email) {
        this.customerId = customerId;
        this.name = name;
        this.email = email;
        this.cart = new ArrayList<>();
    }

    public String getCustomerId() {
        return customerId;
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    public List<Product> getCart() {
        return cart;
    }

    public void addToCart(Product product) {
        cart.add(product);
    }

    public void removeFromCart(Product product) {
        cart.remove(product);
    }

    public void viewCart() {
        System.out.println("Cart contents:");
        for (Product product : cart) {
            System.out.println(product);
        }
    }

    public void checkout() {
        System.out.println("Checking out...");
        viewCart();
        cart.clear();
        System.out.println("Cart is now empty.");
    }
}
```

Step 3: Implement the Order Class

```java
import java.time.LocalDateTime;
import java.util.List;

public class Order {
    private String orderId;
    private Customer customer;
    private List<Product> products;
    private double totalAmount;
    private LocalDateTime orderDate;
```

```
public Order(String orderId, Customer customer, List<Product> products) {
    this.orderId = orderId;
    this.customer = customer;
    this.products = products;
    this.orderDate = LocalDateTime.now();
    this.totalAmount = calculateTotalAmount();
}

public String getOrderId() {
    return orderId;
}

public Customer getCustomer() {
    return customer;
}

public List<Product> getProducts() {
    return products;
}

public double getTotalAmount() {
    return totalAmount;
}

public LocalDateTime getOrderDate() {
    return orderDate;
}

public double calculateTotalAmount() {
    double total = 0;
```

```
        for (Product product : products) {
            total += product.getPrice();
        }
        return total;
    }

    @Override
    public String toString() {
        return "Order{" +
                "orderId='" + orderId + '\'' +
                ", customer=" + customer +
                ", products=" + products +
                ", totalAmount=" + totalAmount +
                ", orderDate=" + orderDate +
                '}';
    }
}
```

Step 4: Implement the Inventory Class

```java
import java.util.ArrayList;
import java.util.List;

public class Inventory {
    private List<Product> products;

    public Inventory() {
        this.products = new ArrayList<>();
    }
```

```
public List<Product> getProducts() {
    return products;
}

public void addProduct(Product product) {
    products.add(product);
}

public Product getProductById(String productId) {
    for (Product product : products) {
        if (product.getProductId().equals(productId)) {
            return product;
        }
    }
    return null;
}

public void updateProductStock(String productId, int quantity) {
    Product product = getProductById(productId);
    if (product != null) {
        product.updateStockQuantity(quantity);
    }
}

@Override
public String toString() {
    return "Inventory{" +
            "products=" + products +
            '}';
}
```

```
}
```

Step 5: Develop the Main Class to Test the System

```java
public class Main {
    public static void main(String[] args) {
        // Create instances of Inventory and Customer
        Inventory inventory = new Inventory();
        Customer customer = new Customer("C001", "John Doe", "john@example.com");

        // Add products to the inventory
        Product product1 = new Product("P001", "Laptop", 999.99, 10);
        Product product2 = new Product("P002", "Smartphone", 499.99, 20);
        inventory.addProduct(product1);
        inventory.addProduct(product2);

        // Simulate adding products to the customer's cart
        customer.addToCart(product1);
        customer.addToCart(product2);

        // View cart contents
        customer.viewCart();

        // Checkout
        customer.checkout();

        // Create an order
        Order order = new Order("O001", customer, customer.getCart());
        // Display order details
        System.out.println(order);

        // Update product stock in inventory after checkout
        inventory.updateProductStock("P001", -1);
        inventory.updateProductStock("P002", -1);

        // Display updated inventory
        System.out.println(inventory);
    }
}
```

This code provides a basic implementation of the online shopping cart system as per the requirements. You can further expand and enhance it with additional features and error handling as needed.