**Question:**

You are tasked with implementing a simple inventory management system for a retail store. Design and implement the following classes:

**Product:**

Create an abstract class called Product with the following attributes:

productId (int): representing the unique identifier of the product.

productName (String): representing the name of the product.

price (double): representing the price of the product.

Implement a parameterized constructor to initialize these attributes.

Include getter methods for each attribute.

Implement an abstract method called calculateDiscount() that should be overridden by subclasses.

**Electronics:**

Create a subclass called Electronics that extends Product.

Add an additional attribute:

warrantyPeriod (int): representing the warranty period in months.

Implement a parameterized constructor to initialize the attributes.

Override the calculateDiscount() method to apply a discount based on the warranty period.

**Clothing:**

Create another subclass called Clothing that extends Product.

Add an additional attribute:

size (String): representing the size of the clothing item.

Implement a parameterized constructor to initialize the attributes.

Override the calculateDiscount() method to apply a discount based on the size (e.g., small size gets a higher discount).

**Inventory:**

Create a class called Inventory that manages a collection of Product objects.

Include methods to add a product to the inventory, remove a product, and display the current inventory.

**Main:**

Create a Main class with a main method to test your implementation.

Create instances of Electronics and Clothing, add them to the Inventory, and display the current inventory.

Demonstrate the use of polymorphism by calling the calculateDiscount method on different types of products.

Ensure that your implementation demonstrates the use of various Java concepts, such as inheritance, abstract classes, interfaces, and the Java Collections Framework for managing the inventory.