Exercise 8-3 Use the abstract and final keywords

In this exercise, you'll change the Product class in the Product application to an abstract class to see how that works, and you'll add an abstract method and implement it in the Book, Software, and CompactDisc subclasses. Then, you'll change the Book class to a final class to see that a final class can't be inherited, and you'll create a final method to see that it can't be overridden.

Change the Product class to an abstract class

1. Open the project named ch08_ex3_Product that's in the ex_starts directory. Then, review the code.
2. Add the abstract keyword to the Product class declaration.
3. Open the ProductApp class, and add this statement before the statement that calls the getProduct method:

```
Product pTest = new Product();
```

4. If you're using NetBeans, a syntax error should be displayed indicating that the Product class is declared as abstract and cannot be instantiated. If this error isn't displayed, save or compile the ProductApp class so it is displayed.
5. Delete the statement you just added. Then, run the application to make sure it works.

Add an abstract method to the Product class

6. Add an abstract method named getDisplayText to the Product class. This method should accept no parameters, and it should return a String object. Then, compile this class.
7. Rename the toString methods in the Book and Software classes to getDisplayText.
8. Modify the ProductApp class so it calls the getDisplayText method of a product instead of the toString method. Then, run the application to be sure it works correctly.

Change the Book class to a final class

9. Add the final keyword to the Book class declaration.
10. Create a class named UsedBook that inherits the Book class. You don't need to include any code in the body of this class. If you're using NetBeans, a syntax error should be displayed indicating that the Book class can't be inherited because that class is final. If this error isn't displayed, save or compile the Book and UsedBook classes so it is displayed.

Add a final method

11. Remove the final keyword from the Book class declaration. Then, add the final keyword to the getDisplayText method of the Book class.
12. Add a getDisplayText method to the UsedBook class to override the getDisplayText method of the Book class. Code this method so it returns an empty string. If you're using NetBeans, a syntax error should be displayed indicating that the getDisplayText method can't be overridden because that method is final. If this message isn't displayed, save or compile the Book and UsedBook classes so it is displayed.
13. Remove the final keyword from the getDisplayText method of the Book class. Now, no syntax errors should be displayed. If you get a warning about the @Override annotation, though, add this annotation to the getDisplayText method of the UsedBook class.