Delegates and Events

Lab Exercise 1: Delegate Basics
Objective: Practice creating and using delegates in C#.
Instructions:

1. Create a delegate named StringModifier that takes a string as input and returns a modified string.
2. Define three methods:
    - Uppercase that converts the input string to uppercase.
    - Lowercase that converts the input string to lowercase.
    - Reverse that reverses the input string.
3. Make each of these methods conform to the StringModifier delegate signature.
4. Create a StringModifier variable and assign one of the methods (e.g., Uppercase) to it.
5. Call the delegate with a string input and observe the output.

Bonus:

- Try chaining multiple delegates together using the + operator. For example, you could combine Uppercase and Reverse to create a delegate that uppercases and then reverses a string.

This exercise will help you understand the basic syntax of delegates and how to assign methods to them.

Lab Exercise 2: Delegate Events
Objective: Simulate a button click event using delegates.
Instructions:

1. Create a class named Button with a public method Click that raises a delegate event named ButtonClicked.
2. Define a ButtonClicked delegate that takes no arguments and returns void.
3. In the Click method, invoke the ButtonClicked delegate if it has any subscribers.
4. Create another class named Form that contains a Button object and a method named SubscribeToButtonClick.
5. Inside SubscribeToButtonClick, add an anonymous method to the ButtonClicked delegate of the Button object. This anonymous method will be executed when the button is clicked.
6. In the Form class, create a button click event handler using the Click method of the Button object.
7. Call the SubscribeToButtonClick method from the button click event handler.
8. Run the program and click the button. You should see the output from the anonymous method subscribed to the ButtonClicked delegate.

This exercise demonstrates how delegates can be used to implement event-driven programming in C#.