### Lab Exercise: Event-Driven Messaging System

#### Objective:

To practice and understand the concepts of Delegates, Anonymous Methods, Lambda Expressions, Multicasting Delegates, Generic Delegates, Events, Observer Pattern, EventHandler, and EventArgs in C#.

#### Scenario:

You are going to create a simple messaging system where users can send and receive messages. This system will use delegates and events to handle message sending and receiving, and will implement the Observer pattern to notify users when a new message is received.

#### Steps:

1. **Create Delegate and EventHandler:**
   * Define a generic delegate MessageReceivedHandler<T>.
   * Define an event MessageReceived using EventHandler<MessageEventArgs>.
2. **Create a User Class:**
   * Define a User class with properties like Name and methods to send and receive messages.
   * Implement the observer pattern where each user can subscribe to the message received event.
3. **Implement Anonymous Method and Lambda Expression:**
   * Use an anonymous method to handle the MessageReceived event for one user.
   * Use a lambda expression to handle the MessageReceived event for another user.
4. **Multicasting Delegates:**
   * Implement a method to demonstrate multicasting delegates where multiple methods are called when a single event is triggered.
5. **Generic Delegates:**
   * Use a generic delegate for a method that processes messages.
6. **Event and Observer Pattern:**
   * Implement the event and observer pattern to notify users when a new message is received.
7. **Main Program:**
   * Create instances of the User class.
   * Subscribe to the MessageReceived event using both anonymous methods and lambda expressions.
   * Send messages and observe the output.