Module 5

What to learn

Event
Delegate
Func Delegate
Lambda

Practice Exercise

Practice 1

do the hands on video provides

Practice 2

Event

Implement a custom event BookAdded in a library system that triggers whenever a new book is added. Display the book details when the event is triggered.

Create a program where an event is raised when a stock price crosses a predefined threshold. Notify all subscribers of the change.

Develop a Timer class that raises an event every second. Create a subscriber class that listens to this event and displays the current time.

Write a program where a BankAccount class raises an InsufficientFunds event when a withdrawal exceeds the balance. Implement multiple subscribers to log the event and alert the user.

Implement an event system for a Door class where an event DoorOpened is raised when the door state changes. Have different subscribers respond to the event (e.g., log the state, notify security).

Practice 3

Delegate

Write a program using a delegate to sort a list of integers in both ascending and descending order. The order should be decided dynamically based on the delegate passed.

Create a Calculator class that performs addition, subtraction, multiplication, and division using delegates. Allow the user to select the operation at runtime.

Implement a delegate in a FileProcessor class to allow dynamic filtering of file names based on file extensions.

Create a NotificationSystem class that sends different types of notifications (email, SMS, push notification) using a delegate.

Write a program where a delegate is used to perform different transformations (e.g., capitalize, reverse) on a string.

Practice 4

Func Delegate

Implement a program using Func<T, TResult> to calculate the area of different shapes (circle, rectangle, triangle) based on user input.

Create a dictionary of operations (add, subtract, multiply, divide) implemented as Func<int, int, int> delegates. Allow the user to choose an operation and provide input.

Use Func<int, bool> to filter a list of numbers for prime numbers and display the result.

Write a program using Func<string, string> to transform a list of strings (e.g., convert to uppercase, trim whitespace, append a suffix).

Implement a method that takes a Func<int, int, bool> to compare two integers and return true if the condition (e.g., greater than, equal) is satisfied.

Practice 5

Lambda

Write a program to find all even numbers in a list using a lambda expression with List<T>.Where().

Use a lambda expression to sort a list of strings by their lengths.

Implement a method that takes a list of integers and a lambda function to filter numbers based on a condition (e.g., greater than 10, divisible by 3).

Create a dictionary of products and prices. Use a lambda expression to find the most expensive product.

Write a program to group a list of words by their first letter using a lambda expression and GroupBy().

Assignment Exercise

Assignment 1

Compute area of rectangle using func delegate

Assignment 2

Compute add of two number using lambda expression

Assignment 3

Comprehensive Question: Event, Delegate, Func Delegate, and Lambda Integration

Scenario:
You are building a **task scheduling system** for a productivity application. The system manages tasks, notifies users of task deadlines, and provides customizable filters for viewing tasks.

Objective:

Implement a **TaskManager** application that integrates **events**, **delegates**, **Func delegates**, and **lambda expressions**.

Requirements:

1. Task Management Using Delegates

Create a Task class with the following properties:

- TaskId (int)
- Title (string)
- Deadline (DateTime)
- Priority (enum: Low, Medium, High)
- IsCompleted (bool)

Implement a TaskManager class with the following methods:

- **AddTask**: Add a task to the list.
- **MarkTaskCompleted**: Use a delegate to update the IsCompleted property of a task based on its TaskId.
- **SortTasks**: Sort tasks by a user-specified criterion (e.g., Deadline, Priority) using a delegate.

2. Notification Using Events

- Add an event TaskDeadlineApproaching in the TaskManager class, triggered when a task's deadline is within 24 hours.
- Create a NotificationService class that subscribes to this event and displays a message with task details.

3. Filtering Tasks Using Func Delegates

- Implement a method FilterTasks in TaskManager that takes a Func<Task, bool> to filter tasks dynamically. Examples:
  - Get all tasks with high priority.
  - Get all incomplete tasks.
  - Get all tasks due within a week.

4. Lambda Expressions for Custom Operations

- Use lambda expressions to:
  - Sort tasks by title length.
  - Find tasks that contain a specific keyword in the title.

```
Task Manager System
--------------------
1. Add a New Task
2. Mark a Task as Completed
3. View All Tasks
4. Filter Tasks by Custom Criteria
5. Trigger Deadline Notifications
6. Exit
```

Online Reference

No online Reference

.NET Core Web API

WEB API (old)

Authentication And Authorization (WEBAPI)(old)

FullStackDevelopment_WIth_Dotnet_AND_Angular