

C# Delegates, Lambda, Events, LINQ

Objective:

The objective of this lab is to equip you with a practical understanding of key C# programming concepts, including delegates, lambda expressions, events, and LINQ (Language Integrated Query).

Scenario:

You are tasked with developing a Smart Home Management System that allows users to efficiently manage various smart devices in their homes. As smart technology becomes increasingly prevalent, this system aims to provide users with a seamless way to control their devices and monitor their home environment. The system will include features that ensure energy efficiency, enhance comfort, and improve security. Users will have the ability to control lights, thermostats, and security cameras through a central application. The system will also enable automation and notifications, allowing users to receive alerts based on device status changes.

Key Features:

1. **Maintain a List of Smart Devices:** The system will keep an updated list of all smart devices, which will include essential information such as device name, type, current status (on/off), temperature (for thermostats), and security status (for cameras). This will provide users with a clear overview of their smart home ecosystem.
2. **Filter Devices Using LINQ:** Users will be able to filter devices based on various criteria, such as device type (e.g., lights, thermostats) and their current operational status (on/off). This feature will allow users to quickly find and manage devices based on their needs.
3. **Use Delegates and Lambda Expressions:** The implementation will use delegates to define callback methods for operations on smart devices. Lambda expressions will provide a concise way to filter and process collections of devices, making the code more readable and efficient.
4. **Trigger Events on Status Changes:** The system will monitor each device for status changes. For example, if a light is turned on or if a thermostat exceeds a specified temperature, an event will be triggered. This event-driven architecture will enable the application to respond promptly to changes in the smart home environment.
5. **Provide Notifications for Device Status Changes:** Users will be able to subscribe to notifications for when the status of specific devices changes. For instance, they can receive alerts when a light is turned on or when a thermostat's temperature exceeds a certain threshold. This feature will keep users informed and help them take timely actions to maintain comfort and security in their homes.

SE 410 - Software Framework Applications

LAB 2

Tasks:

1. **Define a delegate** named DeviceStatusChange to notify subscribers when a device's status changes.
2. **Use a lambda expression** to filter and list all devices that are currently turned on.
3. **Implement a LINQ query** to find all thermostats that are set above a specified temperature threshold.
4. **Create an event** that triggers when a device's status changes, notifying all subscribers.
5. **Write code to subscribe to the status change event** and display a message when it is triggered.