

C# Assignments on Anonymous Type, Delegates, Events and Lambda – Part 7

Assignment 1. Anonymous Types

Question:

Create an anonymous type to represent a book with properties for Title, Author, and Price. Display the details of the book.

Assignment 2. Var and Dynamic

Question:

1. Declare a var variable to hold a list of integers, add some numbers to it, and print them.
2. Declare a dynamic variable, assign different types of values to it, and observe the behavior.

Assignment 3. Delegates

Question:

Create a simple calculator using delegates. Define a delegate `OperationDelegate` that can represent different arithmetic operations (addition, subtraction, multiplication, division). Implement methods for each operation and allow the user to select an operation to perform on two input numbers.

Requirements:

1. Define a delegate called `OperationDelegate` that takes two double parameters and returns a double result.
2. Implement methods for addition, subtraction, multiplication, and division.
3. Use the delegate to call the selected method.
4. Allow the user to input two numbers and choose an operation.
5. Display the result of the operation.

Assignment 4. Events

Question:

Create an event called `OnCompleted` in a class `Process`. Raise the event when a process is completed. Subscribe to the event from the main program to display a message.

Assignment 5. Anonymous Methods

Question:

Write a delegate `GreetDelegate` that takes a string parameter. Assign an anonymous method to the delegate that prints a greeting message.

Assignment 6. Lambda Expressions**Question:**

Write a lambda expression that takes two integers and returns their product. Use `Func<int, int, int>` for the delegate type.

Assignment 7. Expression Tree**Question:**

Create an expression tree for a lambda expression that adds two integers. Compile and execute the expression.

Assignment 8. Using Delegates to Hold Methods with the Same Signature**Question:**

Create a delegate called `StringOperationDelegate` that accepts a string parameter and returns a string. Write three different methods that modify the string in different ways (e.g., convert to uppercase, reverse, add a prefix). Use the delegate to call each method.

Assignment 9. Lambda Expressions to Filter and Process Lists**Question:**

Given a list of products with properties `Name` and `Price`, use lambda expressions to:

1. Filter the list to find products that cost more than \$50.
2. Sort the list by price in descending order.
3. Select and display only the names of products from the filtered list.

Assignment 10. Delegate as a Callback Mechanism**Question:**

Create a delegate called `CalculationDelegate` that takes two integers and returns an integer. Implement methods for addition and subtraction, and write a method `Calculate` that accepts two integers and a delegate as parameters. Use the delegate to perform addition and subtraction operations by passing different methods.

Assignment 11. Introduction to Asynchronous Programming**Question:**

Write a C# program that simulates a long-running operation using `Task.Delay()` and runs asynchronously. Display a message before and after the operation completes.

Assignment 12. Async and Await Keywords

Question:

Create a method that downloads data from a sample URL asynchronously. Use `HttpClient` to fetch the data and print the length of the content. The method should use `async` and `await` keywords.

Assignment 13. Task and Task<T>

Question:

Write a C# program that calculates the sum of an array of integers asynchronously using `Task<int>`. Return the result from the asynchronous method and display it.

Assignment 14. Handling Exceptions in Asynchronous Methods

Question:

Modify the previous example to handle exceptions that might occur during the sum calculation. Catch the exception in the `Main` method and display an error message.

Assignment 15. Parallel Programming with Task Parallel Library (TPL)

Question:

Create a program that runs multiple tasks in parallel to simulate multiple operations, such as calculating the factorial of different numbers. Print the result of each task when completed.

Assignment 16. LINQ Query Syntax and Method Syntax

Question: Write a LINQ query using both query syntax and method syntax to retrieve all numbers greater than 5 from a list of integers.

Assignment 17. LINQ Operators (Select, Where, GroupBy, Join, etc.)

Question 1: Given a list of strings representing student names, write a LINQ query that selects the names starting with the letter 'A' and orders them alphabetically.

Assignment 18. LINQ to Objects

Question: Write a LINQ query to find the sum of all even numbers in an array of integers.

Assignment 19. LINQ to XML

Question: Create an XML document representing a list of books, and use LINQ to XML to query the titles of books published after 2015.

Assignment 20. LINQ to SQL

Question: Given a database table Employees with columns ID, Name, and Department, write a LINQ to SQL query to select all employees from the "IT" department.