Session: 15

Advanced Concepts of C#

Objectives

- Describe system-defined generic delegates
- Define lambda expressions
- Explain query expressions LinQ
- Describe Windows Communication Framework (WCF)
- Explain parallel programming
- Explain dynamic programming

System-Defined Generic Delegates

- Following are the commonly used predefined generic delegates:

`Func<TResult>()` Delegate	`Func<T, TResult>(T arg)` Delegate	`Func<T1, T2, TResult>(T1 arg1, T2 arg2)` Delegate
represents a method having zero parameters and returns a value of type `TResult`.	represents a method having one parameter of type T and returns a value of type `TResult`.	represents a method having two parameters of type `T1` and `T2` respectively and returns a value of type `TResult`.

```
public class WordLength{
    public static void Main() {
        Func<string, int> cntWord = Count;
        string location = "Netherlands";
        // Use delegate instance to call Count method
        Console.WriteLine("The number of characters in the input is:
        {0} ",cntword(location).ToString());
    }
    private static int Count(string inputString) {
        return inputString.Length;
    }
}
```

- A method associated with a delegate is never invoked by itself, instead, it is only invoked through the delegate.

Syntax

```
parameter-list => expression or statements
```

- where,
  - **parameter-list** : is an explicitly typed or implicitly typed parameter list
  - => : is the lambda operator

Example

```
class Program {
    delegate int ProcessNumber(int input);
    static void Main(string[] args) {
        ProcessNumber square = n => n * n;
        Console.WriteLine(square(5));
    }
}
```

Lambdas with Standard Query Operators

- Lambda expressions can also be used with standard query operators.

Operator	Description
Sum	Calculates sum of the elements in the expression
Count	Counts the number of elements in the expression
OrderBy	Sorts the elements in the expression
Contains	Determines if a given value is present in the expression

- Example

```
public class NameSort{
    public static void Main() {
       // Declare and initialize an array of strings
        string[]names={"Hanna","Jim","Peter","Karl","Abby”};
        Foreach (string item in names.OrderBy(s => s)) {
                Console.WriteLine(item);
       }
    }
}
```

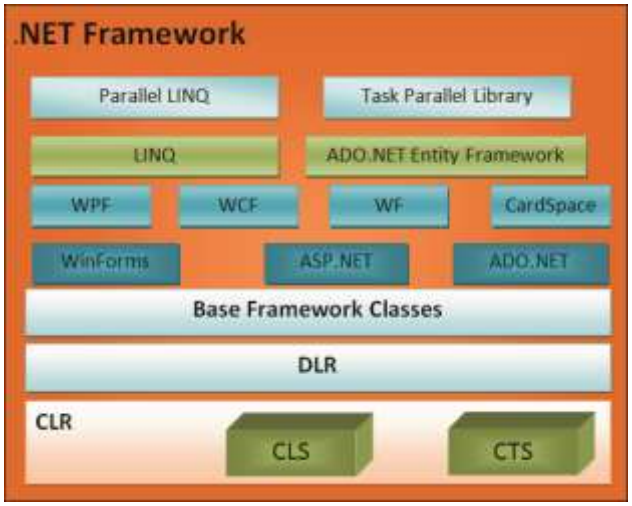
LINQ - Query Expressions

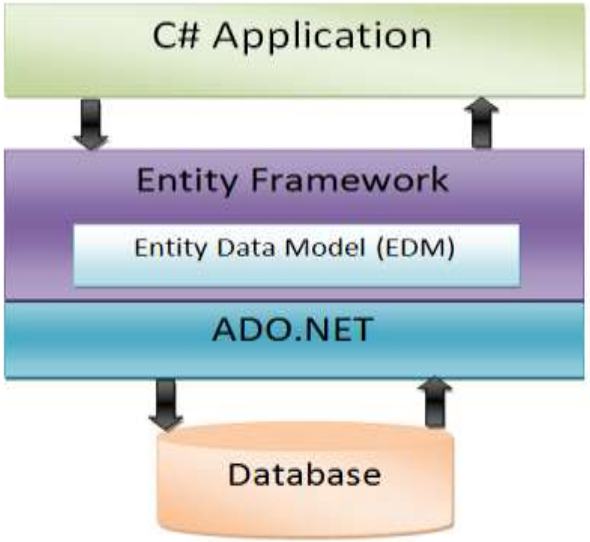
- A query expression is a query that is written in syntax using clauses such as **from**, **select**… These clauses are an inherent part of a LINQ query.
- LINQ is introduced in Visual Studio 2008 that simplifies working with data present in various formats in different data sources.
- A **from** clause must be used to start a query expression and a **select** or **group** clause must be used to end the query expression.

Clause
from
where
select
group
orderby
ascending
descending

```
class Program  {
   static void Main(string[] args) {
       string[] names = {"Hanna","Jim","Pearl","Mel",
          "Jill","Peter","Karl","Abby","Benjamin" };
       IEnumerable<string> items = from word in names
                   where word.EndsWith("l")
                    select word;
       foreach (string s in items){
           Console.WriteLine(s);
       }
   }
}
```

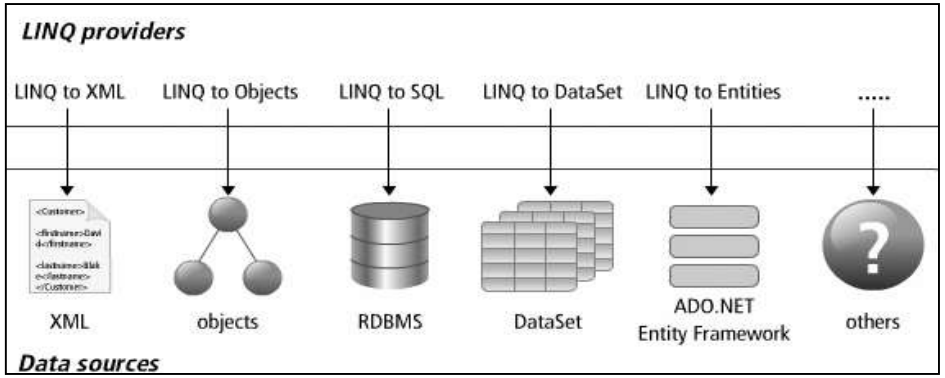
- The Entity Framework is an implementation of the Entity Data Model (EDM), which is a conceptual model that describes the entities and the associations they participate in an application.
- EDM allows to handle data access logic by programming against entities without having to worry about the structure of the underlying data store and how to connect with it.



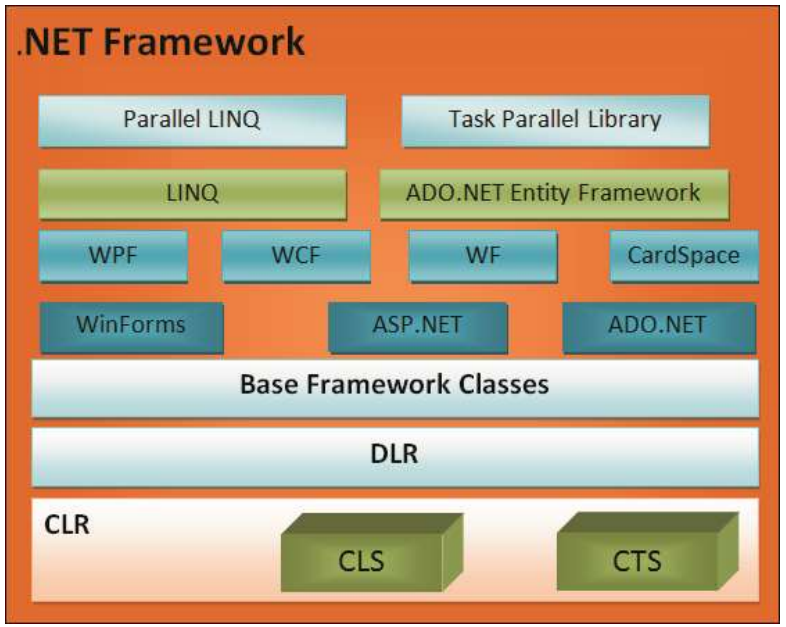


Querying Data by Using LINQ Query Expressions

- LINQ provides a consistent programming model to create standard query expression syntax to query different types of data sources.
- However, different data sources accept queries in different formats. To solve this problem, LINQ provides the various LINQ providers, such as LINQ to Entities, LINQ to SQL, LINQ to Objects, and LINQ to XML…
- To create a query, needs a data source against which the query will execute. An instance of the ObjectQuery class represents the data source.
- In LINQ to entities, a query is stored in a variable. When the query is executed, it is first converted into a command tree representation that is compatible with the Entity Framework.
- Then, the Entity Framework executes the query against the data source and returns the result.



- Windows Communication Foundation (WCF) is a framework for creating loosely-coupled distributed application based on Service Oriented Architecture (SOA).
- SOA is an extension of distributed computing based on the request/ response design pattern.
- SOA allows creating interoperable services that can be accessed from heterogeneous systems.
- Interoperability enables a service provider to host a service on any hardware or software platform that can be different from the platform on the consumer end.



The System.Threading.Thread Class

- allows to create and control a thread in a multithreaded application.
- Each thread passes through different states that are represented by the members of the **ThreadState** enumeration.
- A new thread can be instantiated by passing to the constructor of the Thread class a **ThreadStart** delegate that represents the method that the new thread will execute.
- Once a thread is instantiated, it can be started by making a call to the Start() method.

Snippet

```
class ThreadDemo {
   public static void Print() {
       while (true)  Console.Write("1");
   }
   static void Main (string [] args) {
       Thread newThread = new Thread(new ThreadStart(Print));
       newThread.Start();
       while (true)  Console.Write("2");
   }
}
```

Concurrent Collections

- The generic collection classes provides improved type safety and performance. However, they are not thread safe.
- To address the problems, the .NET Framework provides concurrent collection classes in the System.Collections.Concurrent namespace.
- These classes relieves programmers from providing thread synchronization code when multiple threads simultaneously accesses these collections.
- The important classes :

ConcurrentDictionary <TKey, TValue>	• Is a thread-safe implementation of a dictionary of key-value pairs.
ConcurrentQueue<T>	• Is a thread-safe implementation of a queue.
ConcurrentStack<T>	• Is a thread-safe implementation of a stack.
ConcurrentBag<T>	• Is a thread-safe implementation of an unordered collection of elements.

The System.Threading.Task Class

- TPL (`Task Parallel Library`) provides the Task class represents an asynchronous task in a program.
- Task is created by providing a user delegate that encapsulates the code that the task will execute. The delegate can be a named delegate, such as the Action delegate, an anonymous method, or a lambda expression. After that, calls the Start() method to start the task.
- This method passes the task to the task scheduler that assigns threads to perform the work.
- To ensure that a task completes before the main thread exits, call the Wait() method of the Task class.
- To ensure that all the ta
array of the Tasks object
- The Task class also provi
operation.

Snippet

```
class TaskDemo {
    private static void printMessage()  {
        Console.WriteLine("Executed by a Task");
    }

    static void Main (string [] args) {
        Task t1 = new Task(new Action(printMessage));
        t1.Start();
        Task t2 = Task.Run(() => printMessage());
        t1.Wait();
        t2.Wait();
        Console.WriteLine("Exiting main method");
    }
}
```

Asynchronous Methods

- TPL provides support for asynchronous programming through two new keywords: `async` and `await`.
- These keywords can be used to asynchronously invoke long running methods in a program.
- A method marked with the `async` keyword notifies the compiler that the method will contain at least one await keyword.
- If the compiler finds a method marked as `async` but without an `await` keyword, it reports a compilation error.
- The `await` keyword is applied to an operation to temporarily stop the execution of the `async` method until the operation completes.
- In the meantime, control returns to the `async` method's caller. Once the operation marked with `await` completes, execution resumes in the `async` method.
- A method marked with the `async` keyword can have either one of the following return types:
  - `void`
  - `Task`
  - `Task<TResult>`

Parallel LINQ (PLINQ)

- LINQ to Object refers to the use of LINQ queries with enumerable collections, such as List<T> or arrays.
- PLINQ is the parallel implementation of LINQ to Object. While LINQ to Object sequentially accesses an in-memory IEnumerable or IEnumerable<T> data source, PLINQ attempts parallel access to the data source based on the number of processor in the host computer.
- For parallel access, PLINQ partitions the data source into segments, and then executes each segment through separate threads in parallel.
- The System.Linq.ParallelEnumerable provides methods that implement PLINQ functionality.

```
string[] arr = new string[] { "Peter‟,"Sam", Philip","Andy", "Philip",
"Mary", "John","Pamela"};
var query = from string name in arr select name;
Console.WriteLine("Names retrieved using sequential LINQ");
foreach (var n in query) {
    Console.WriteLine(n);
}

var plinqQuery = from string name in arr.AsParallel() select name;
Console.WriteLine("Names retrieved using PLINQ");
foreach (var n in plinqQuery)      {
    Console.WriteLine(n);
}
```

Dynamic Programming

- C# provides dynamic types to support dynamic programming for interoperability of .NET applications with dynamic languages such as IronPython and COM APIs such as the Office Automation APIs.
- The C# compiler does not perform static type checking on objects of a dynamic type. The type of a dynamic object is resolved at runtime using the Dynamic Language Runtime (DLR).
- A programmer using a dynamic type is not required to determine the source of the object's value during application development.
- However, any error that escapes compilation checks causes a run-time exception.

Snippet

```
class DemoClass {
  public void Operation(String name)    {
    Console.WriteLine("Hello {0}", name);
  }
}
class DynamicDemo {
  static void Main(string[] args)    {
    dynamic dynaObj = new DemoClass();
    dynaObj.Operation();
  }
 }
```

Summary

- System-defined generic delegates take a number of parameters of specific types and return values of another type.
- A lambda expression is an inline expression or statement block having a compact syntax and can be used in place of a delegate or anonymous method.
- A query expression is a query that is written in query syntax using clauses such as from, select, and so forth.
- The Entity Framework is an implementation of the Entity Data Model (EDM), which is a conceptual model that describes the entities and the associations they participate in an application.
- WCF is a framework for creating loosely-coupled distributed application based on Service Oriented Architecture (SOA).
- Various classes and interfaces in the `System.Threading` namespace provide built-in support for multithreaded programming in the .NET Framework.
- To make parallel and concurrent programming simpler, the .NET Framework introduced TPL, which is a set of public types and APIs in the `System.Threading` and `System.Threading.Tasks` namespaces.