**MODULE - 1 [ Getting started with MVC .Net]**

1. **What is role of client?**

* A **client** is a computer hardware device or software that accesses a service made available by a server. The server is often (but not always) located on a separate physical computer.

1. **What is role of server?**

* A Server is a program or a device that provides functionality for called clients which are other programs or devices. This architecture is called the client-server model.
* A single overall computation is distributed across multiple processes or devices. Servers can provide various functionalities called services. These services include sharing data or resources among multiple clients or performing computations for a client. Multiple clients can be served by a single server, and a single client can use multiple servers

1. **What is client server architecture?**

* Client-server architecture is a computing model in which the server hosts, delivers, and manages most of the resources and services requested by the client. It is also known as the networking computing model or client-server network as all requests and services are delivered over a network. The client-server architecture or model has other systems connected over a network where resources are shared among the different computers.

1. **What is role of compiler?**

* A Compiler is a software that typically takes a high level language (Like C++ and Java) code as input and converts the input to a lower level language.

1. **What is difference between Compiler and Interpreter?**

* **Compiler:**

The Compiler is a translator which takes input i.e., High-Level Language, and produces an output of low-level language i.e. machine or assembly language. The work of a Compiler is to transform the codes written in the programming language into machine code (format of 0s and 1s) so that computers can understand.

* **Interpreter:**

An Interpreter is a program that translates a programming language into a comprehensible language. The interpreter converts high-level language to an intermediate language. It contains pre-compiled code, source code, etc.

1. **What is MVC?**

* The Model-View-Controller (MVC) framework is an architectural/design pattern that separates an application into three main logical components Model, View, and Controller. Each architectural component is built to handle specific development aspects of an application. It isolates the business logic and presentation layer from each other. It was traditionally used for desktop graphical user interfaces (GUIs). Nowadays, MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects. It is also used for designing mobile apps.

1. **What is communication protocol and what is difference between**

**HTTP and HTTPS?**

* **HyperText Transfer Protocol (HTTP):** HyperText Transfer Protocol (HTTP) is a protocol using which hypertext is transferred over the Web. Due to its simplicity, http has been the most widely used protocol for data transfer over the Web but the data (i.e. hypertext) exchanged using http isn’t as secure as we would like it to be. In fact, hyper-text exchanged using http goes as plain text i.e. anyone between the browser and server can read it relatively easily if one intercepts this exchange of data.
* **Hypertext Transfer Protocol Secure (HTTPS):** Hypertext Transfer Protocol Secure (HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication. In HTTPS, the communication protocol is encrypted using Transport Layer Security.

1. **What is .net?**

* .NET is a free, cross-platform, open source developer platform for building many different types of applications.
* With .NET, you can use multiple languages, editors, and libraries to build for web, mobile, desktop, games, IoT, and more.

1. **What is CLR?**

* The **Common Language Runtime (CLR)** is a component of the Microsoft .NET Framework that manages the execution of .NET applications. It is responsible for loading and executing the code written in various .NET programming languages, including C#, VB.NET, F#, and others.
* When a C# program is compiled, the resulting executable code is in an intermediate language called Common Intermediate Language (CIL) or Microsoft Intermediate Language (MSIL). This code is not machine-specific, and it can run on any platform that has the CLR installed. When the CIL code is executed, the CLR compiles it into machine code that can be executed by the processor.

1. **What is difference between CLS and CTS?**

* **CLS:** Comman Language Specification. it is Subset of CTS.
* **CTS:** Comman Type systemes. it can have different Types of Datatypes. it can be supports two types.

1) Value Type.

2) Refference type.

* **1) Value type:** It can Allocated Memory in Stack.

Ex: int i =10;

I value goes to stored in stack memory,int,float,struct etc.

* **2) Refference type:** It can Allocated Memory in Heap .

ex: class,object, delegate,interface,string etc.

1. **What do you mean by design pattern?**

* Design patterns represent the best practices used by experienced object-oriented software developers. Design patterns are solutions to general problems that software developers faced during software development. These solutions were obtained by trial and error by numerous software developers over quite a substantial period of time.

1. **What is difference between Asp.Net and MVC.Net?**

* **ASP.NET** is a popular open-source web application framework developed by Microsoft that is used for building dynamic websites and web applications.
* **MVC.NET**, on the other hand, is a more recent addition to the ASP.NET family, and it is a web application framework that implements the Model-View-Controller (MVC) pattern.
* The most significant differences between ASP.NET and MVC.NET lie in their architecture and development model. ASP.NET is based on the event-driven model, while MVC.NET is based on the request-response model. This means that ASP.NET is best suited for applications that require a high degree of interactivity, while MVC.NET is better for applications that require a more modular and scalable architecture.