UNIT DIT503: C#.NET PROGRAMMING

UNIT CODE: L/617/6742

Introducing ASP.NET & C# Fundamentals

Learning Outcomes and Assessment Criteria

1. Understand the use of ASP.NET.

   1.1 Analyse the components / structure of ASP.NET.
   1.2 Evaluate the advantages and disadvantages of using ASP.NET compared with other web development models.
   1.3 Analyse the advantages of using validators.

2. Design web applications using ASP.NET and ADO.NET.

   2.1 Use styles, themes and master pages to create an attractive and easily navigable web applications.
   2.2 Display dynamic data from a relational database by using ADO.NET and data binding through different languages including C#.
   2.3 Create a web page that uses client side navigation, client side browser redirect, cross page posting and server side transfer that meets the brief.

EVOLUTION OF WEB DEVELOPMENT

The rise of the internet and businesses online has driven the evolution of websites. Websites have evolved from static pages to interactive, dynamic, and AI-driven experiences.

Early Web (1990s)

- 1990: Tim Berners-Lee created the first webpage using HTML, HTTP, and a web browser (WorldWideWeb).
- 1991-1995: HTML 1.0 & 2.0 – Basic text-based websites, no styling or interaction.
- 1995: Introduction of CSS 1 and JavaScript for basic styling and interactivity.

Rise of Dynamic Websites (2000s)

- HTML 4 (1999): Allowed better structuring of content.
- CSS 2 (1998): Improved styling and layout control.
- JavaScript & AJAX (2005): Enabled dynamic, real-time content updates (e.g., Google Maps).
- PHP, MySQL, ASP.NET: Backend technologies allowed database-driven websites.

EVOLUTION OF WEB DEVELOPMENT

Responsive & Interactive Web (2010s)

- HTML5 & CSS3 (2014): Improved multimedia support, animations, and mobile responsiveness.
- JavaScript Frameworks: Rise of React, Angular, and Vue.js for interactive applications.
- Mobile-first design: Websites adapted to various screen sizes.

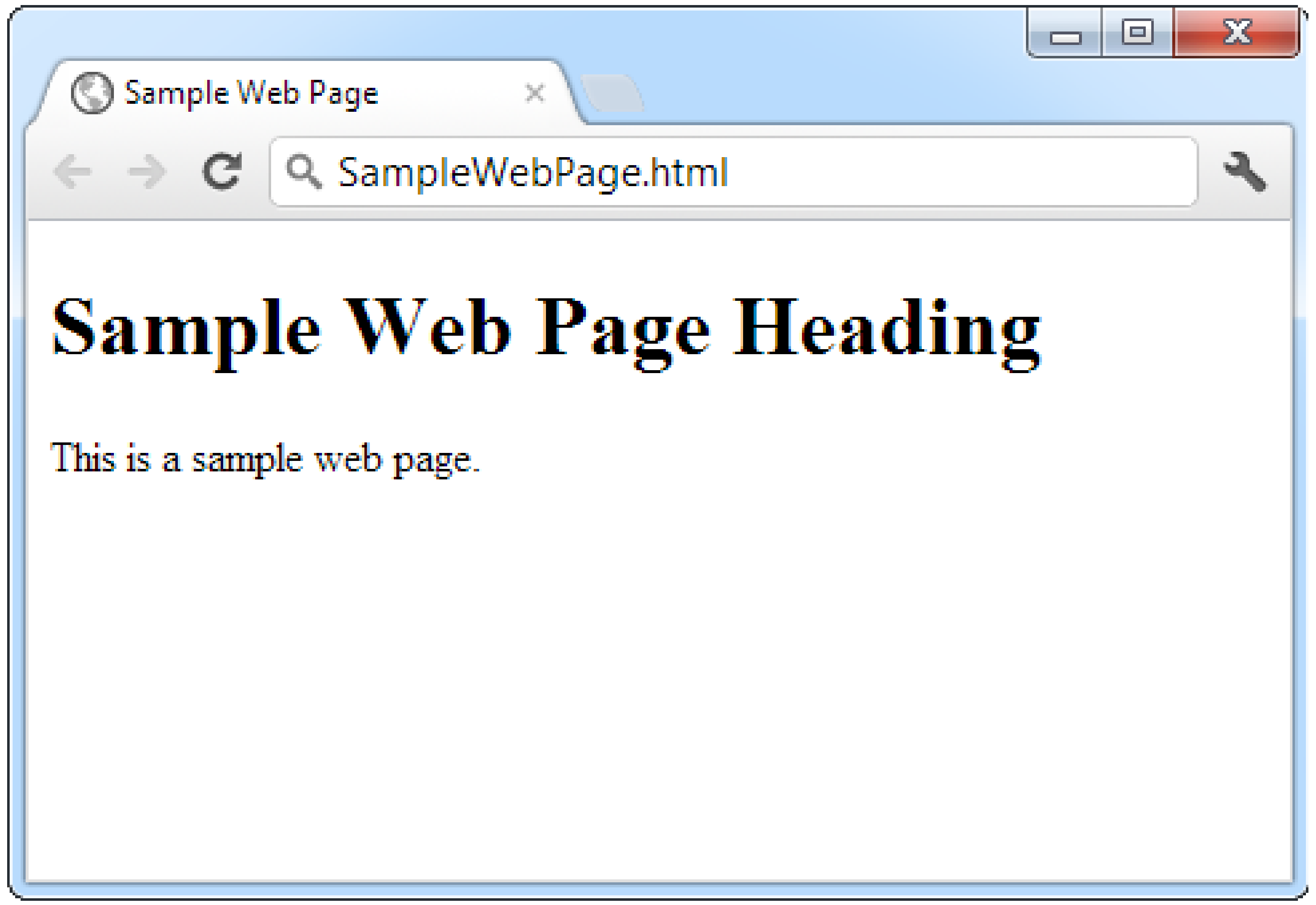
Modern Web (2020s - Present)

- Progressive Web Apps (PWAs): Blending web and mobile experiences.
- Artificial Intelligence & Machine Learning: Personalized recommendations, chatbots.
- Web 3.0: Blockchain, decentralized web, and smart contracts.
- Low-code & No-code platforms: Simplifying web development.

HYPERTEXT MARKUP LANGUAGE

- HTML (HyperText Markup Language) is the foundation of web pages.
- It structures web content using tags and elements.
- Works with CSS (for styling) and JavaScript (for interactivity).

```html
<!DOCTYPE html>
<html>
<head>
   <title>Sample Web Page</title>
</head>
<body>
   <h1>Sample Web Page Heading </h1>
   <p>This is a sample web page.</p>
</body>
</html>
```



Sample Web Page Heading

This is a sample web page.

.NET FRAMEWORK

Overview

The .NET Framework is a comprehensive development platform developed by Microsoft for building and running applications on Windows. The framework facilitates the development of applications that can run on various platforms, including desktops, web servers, and mobile devices.

Design Principles

- Interoperability: Facilitates seamless interaction between new and existing applications.
- Portability: Enables applications to run across various platforms and devices.
- Security: Provides a robust security model to protect applications and data.
- Memory Management: Automates memory allocation and garbage collection, enhancing performance.
- Simplified Deployment: Streamlines application deployment with minimal configuration.

.NET FRAMEWORK KEY COMPONENTS

Common Language Specification (CLS)

The CLS is a set of rules that languages must adhere to in order to be compatible with the .NET Framework. It ensures that code written in one language can be used by other .NET-compliant languages, promoting interoperability.

Assemblies and Metadata

Assemblies are the building blocks of .NET applications, containing compiled code and metadata. They provide information about the types, versioning, and security permissions of the code. Assemblies can be either executable (.exe) or library (.dll) files.

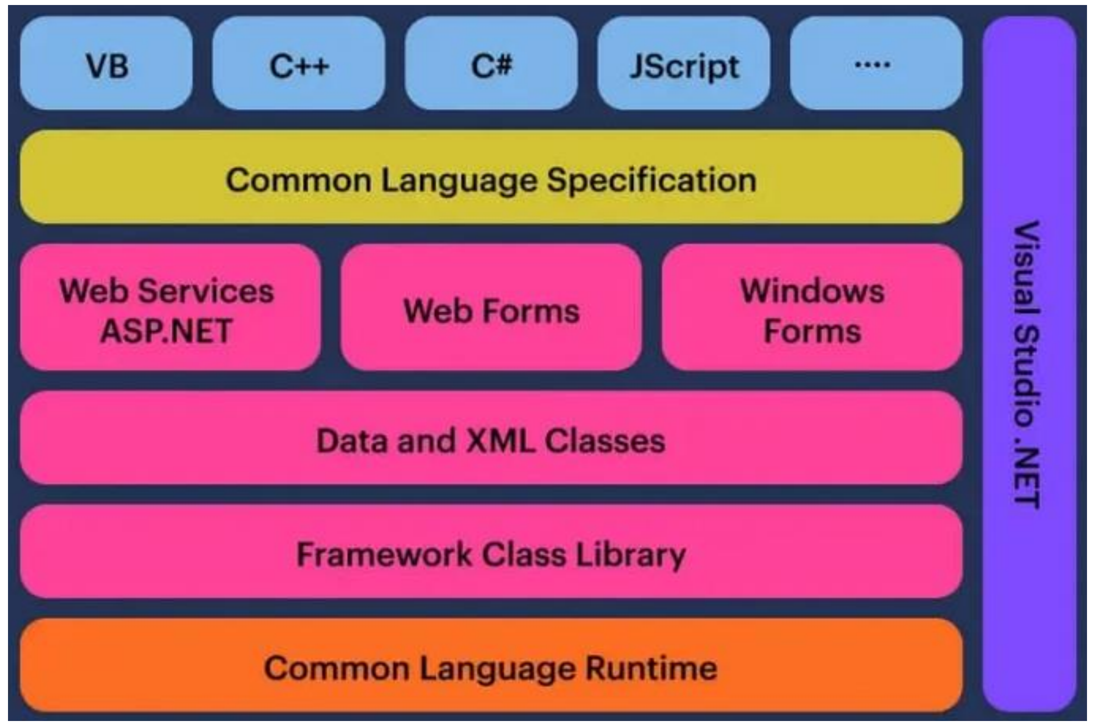
Windows Forms

Windows Forms is a graphical (GUI) class library included in the .NET Framework, providing a platform for developing rich desktop applications with user interfaces. It offers a wide range of controls and components for building Windows-based applications.

Common Language Runtime (CLR)

The CLR is the execution engine for .NET applications, providing essential services such as:

- Memory Management: Handles automatic memory allocation and garbage collection.
- Security Enforcement: Ensures code access security and verification.
- Exception Handling: Manages errors and exceptions during program execution.
- Thread Management: Supports multithreading and parallel processing.
- Just-In-Time (JIT) Compilation: Converts Intermediate Language (IL) code into native machine code at runtime, optimizing performance.



Web Forms

WebForms allows developers to build dynamic web applications using a drag-and-drop, event-driven programming model. WebForms simplifies web development with a design similar to Windows Forms, making it easier for developers familiar with desktop applications to transition to web development.

Framework Class Library (FCL)

The FCL is an extensive collection of reusable classes, interfaces, and value types that provide access to system functionality. Key features include:

- Base Class Library (BCL): Offers fundamental classes for data types, events, exceptions, and more.
- Data and XML Classes: Support data access and XML manipulation.
- Networking Classes: Facilitate network communications and services.
- Windows Forms and WPF: Enable the creation of rich desktop applications.
- ASP.NET: Provides tools for building dynamic web applications and services.

.NET FRAMEWORK

Advantages

- Security: Offers enhanced application security with features like managed code and role-based protection.
- Object-Oriented Programming (OOP): Promotes a modular structure, enhancing code manageability and reuse.
- Ease of Use & Maintenance: Simplifies application configuration, deployment, and device authentication.
- Time Efficiency: Reduces coding requirements, leading to faster development and quicker time-to-market.

Uses of .NET Framework

- Web Applications: Develop dynamic websites and services.
- Desktop Applications: Create rich-featured Windows-based applications.
- Mobile Applications: Build cross-platform mobile apps.
- Gaming: Develop immersive gaming experiences.
- Internet of Things (IoT): Create applications for connected devices.

ASP.NET (ACTIVE SERVER PAGES .NETWORK ENABLED TECHNOLOGIES)

- ASP.NET is a web application framework developed by Microsoft to build dynamic web applications, web services, and APIs.
- It provides a powerful and flexible platform for web development using C#, VB.NET, or F#.
- It is cross-platform and supports Windows, Linux, and macOS.
- ASP.NET uses server-side programming to avoid several problems:

Isolation: Client-side code can't access server-side resources. For example, a client side application has no easy way to read a file or interact with a database on the server (at least not without running into problems with security and browser compatibility).

Security: End users can view client-side code. And once malicious users understand how an application works, they can often tamper with it.

Thin clients: In today's world, web-enabled devices such as tablets and smartphones are everywhere. These devices usually have some sort of built-in web browsing ability, but they may not support client-side programming platforms such as Flash or Silverlight.

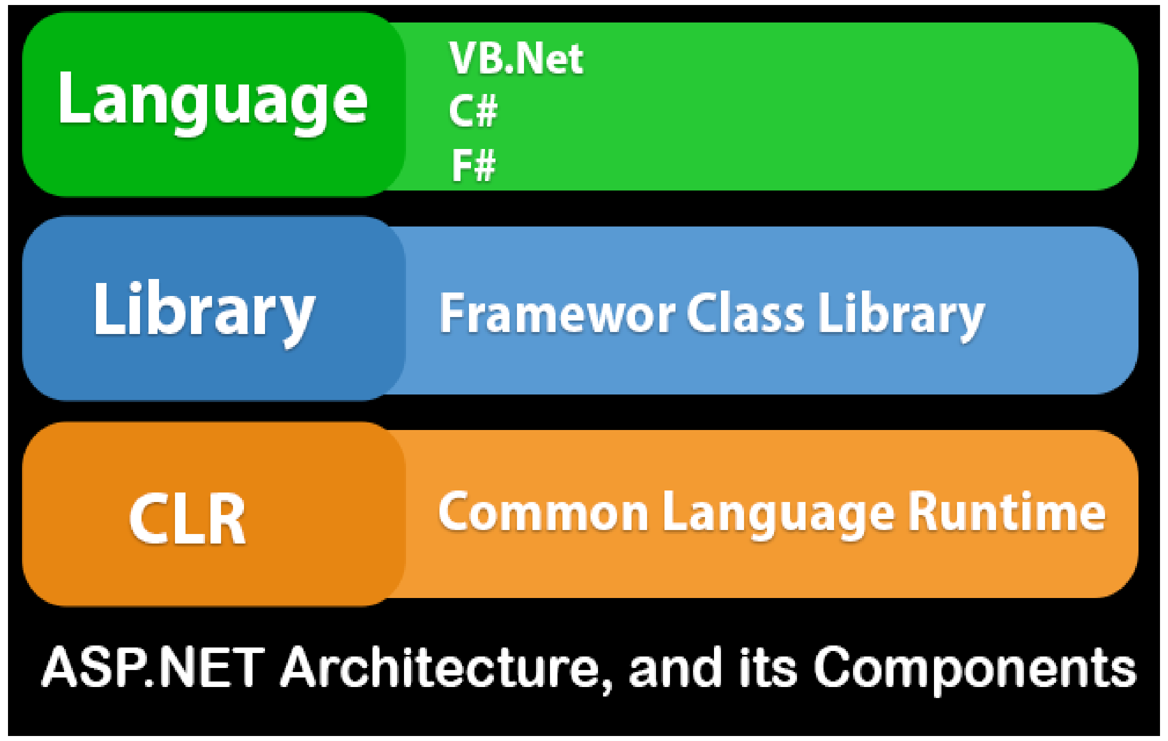
ASP.NET ARCHITECTURE

Language

- ASP.NET applications are written using C#, VB.NET, or F#.
- These languages are compiled into Intermediate Language (IL) by the .NET compiler.

Library (Framework Class Library)

- ASP.NET provides a rich set of pre-built libraries for web development.
- Includes ASP.NET Core libraries, Web API, Entity Framework, and UI libraries.
- Supports built-in security, data access, and web services.



ASP.NET Architecture, and its Components

Common Language Infrastructure (CLI)

- Defines the execution environment for .NET applications.
- Includes:

  Common Language Runtime (CLR)
  → Executes IL code.

  Common Type System (CTS)
  → Ensures type compatibility.

  Garbage Collector (GC)
  → Manages memory allocation.

  Just-In-Time Compiler (JIT)
  → Converts IL code to machine

  code at runtime.

CHARACTERISTICS OF ASP.NET

- **Code-Behind Model:**

  The Code-Behind Model in ASP.NET separates HTML (Presentation Layer) from Server-Side Logic (Business Logic).

- **Directives:**

  Directives provide page-level instructions to the ASP.NET engine.

- **Caching:**

  Caching improves performance by storing frequently accessed data in memory.

- **User Controls:**

  User controls allow developers to create reusable UI components. Similar to custom components in other frameworks. Have .ascx extension.

- **State Management**

  State management is used to preserve data across requests. Since HTTP is stateless, ASP.NET provides mechanisms to maintain user session, data, and controls.

ADVANTAGES OF ASP.NET

Feature	ASP.NET	Other Frameworks
Performance	JIT compilation, caching	Requires optimization
Security	In-built authentication & authorization	External security needed
Scalability	Suited for enterprise-level applications	Limited scalability
Cross-platform	ASP.NET Core supports Linux, Windows, macOS	Some frameworks are OS-specific
Multi-language support	Supports C#, VB.NET, F#	Most frameworks support only one language
Integration	Easy integration with Azure & SQL Server	Manual configuration needed

DISADVANTAGES OF ASP.NET

Challenge	Explanation
Learning Curve	ASP.NET has a steeper learning curve compared to PHP or JavaScript-based frameworks.
Resource-Intensive	ASP.NET applications can be memory-consuming on large servers.
Platform Dependency	Some older versions of ASP.NET are Windows-dependent (before ASP.NET Core).

Q
A

Sriyantha Deepal

(BICT Hons. Specializing Software Technology – USJ)

Mobile: 076 2815533

Email: aasriyanthadeepal@gmail.com