Morning : (90 mins) Getting started with Git basics	
Introduction to Version Control	What is version control and why it is essential? Centralized vs. Distributed version control systems. Git Basics, Branching and Merging working with Remote Repositories
Implementing CI/CD	What is CI/CD and why is it important? ● Continuous integration / Continuous Development ● Faster feedback ● Reduce Bugs ● Automation ● Devops Culture Key concepts: Build, Test, Release, Deploy. Below is a CI/CD lifecycle. Code -> Build -> Test->Release->Deploy Tools for CI/CD: Jenkins -> Enterprise Code Integration GitHub Actions -> Repo native CI/CD GitLab CI, -> tools for CI Azure Devops -> Microsoft ecosystem
Afternoon (90 mins)Introduction & Basic Programming Concepts	
Getting started with .NET platform	Installation of Visual Studio 2019 **Visual Studio 2019 (IDE)** 1. Full features IDE 2. Debugger, intellisense,designer support 3. best for enterprise scale projects **VS Code (light weight IDE)** ● Light weight & faster development ● Crossplatform only ● Requires extensions for C#.NET SDK) Web based tools (One Complier) Introduction to .NET framework What is .NET ? key features of it ● initially NGWS was later renamed to .(DOT) Next generation Technology ● Free, open source, Crossplatform developer platform for building application

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.

	○ Web application (Front End & BackEnd) ○ Desktop application (Windows app) ○ Cloud application ○ Mobile & IOT applications Features : 1. language INteroperability (C#, F# , VB.NET) 25+ language 2. Automatic memory management 3. Rich base Library 4. Cross platform execution (create once reuse on other OS) C -> C with Classes _> Objective C-> C++ → C ++ ++ # is used for Higher nodes What is C# (Sharp) ? - Anders Heilsberg -It is a language used in .NET ecosystem. -Modern, Object-oriented, Strongly typed, Programming language developed by Microsoft. -Primarily it is used for creating : Secure , scalable, high performance application : (Enterprise grade) across for web, windows, desktop, cloud, mobile etc. Current version of C# https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-14 .NET Framework | .NET Core | .NET 6
	.NET framework architecture
	Common type system
	Common language specification
	Common language runtime(IMp)
	Base class library Why Object is the ultimate base class for all types ?
	Introduction to Classes
	Reading and Writing in the console
	Assembly (dll/exe)

Types in C#

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.

	Private
	Shared
	Types
	Value type and Reference type
	Boxing and Unboxing
	Strings : immutable
	String Builder : mutable
	Difference between String and StringBuilder
	Nullable types (defined using ?)

Every Day
Evening Practice : 03:30 -05:30 : 120 mins
MCQ assessment : 25-30 mins
Coding Assessment : Platform Code -Eval /Coding
last 15 mins Summarization for the daily topics

Milestone 1,2,3,4 with passing % is 80
Final Milestone
Capstone project (5Days) Multiple Sprints 1 & 2(100 marks)
Why ?

1. Real markers for your skillset
2. Understanding user stories and implementation of solution
3. It will help in getting your first project.

List of softwares that we need to install to get started :

1. **VS code**
2. **For Azure**

Where C# is used ?

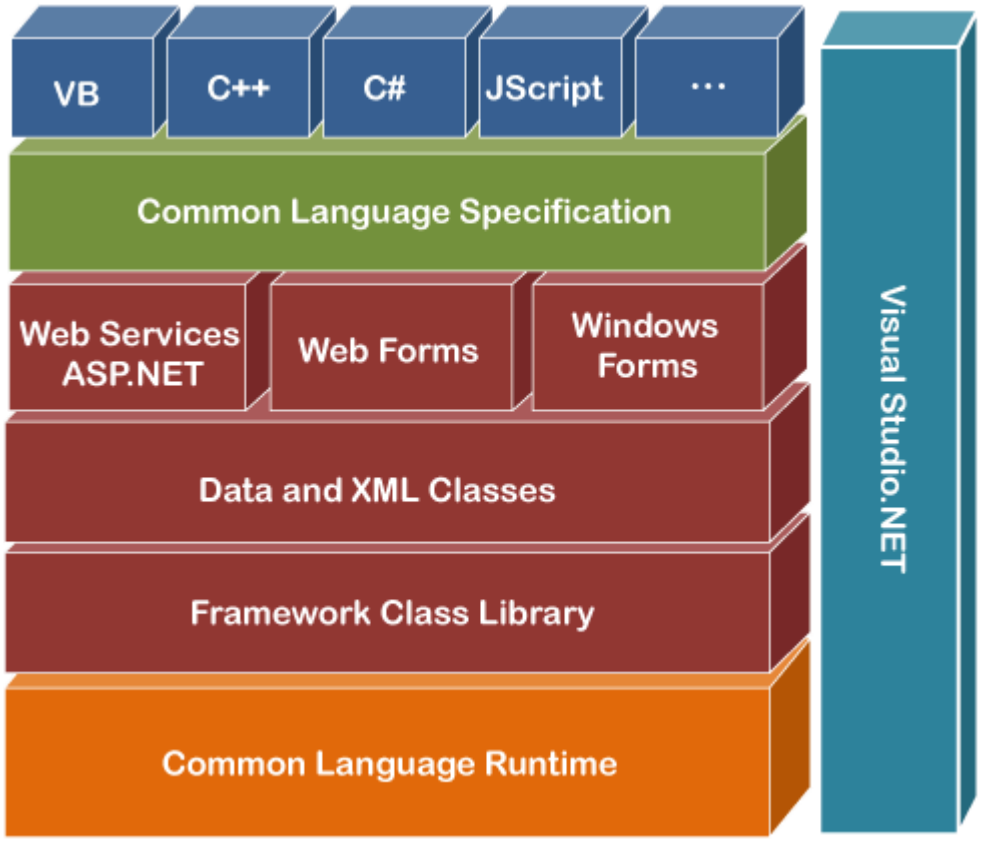
Application Type	Examples
Web Applications & APIs	ASP.NET Core web apps, REST APIs
Enterprise Systems	ERP, CRM, Banking, Insurance platforms
Desktop Applications	Windows Forms, WPF applications
Cloud & Microservices	Azure Functions, microservices
Mobile Applications	Xamarin / .NET MAUI
Game Development	Unity game engine

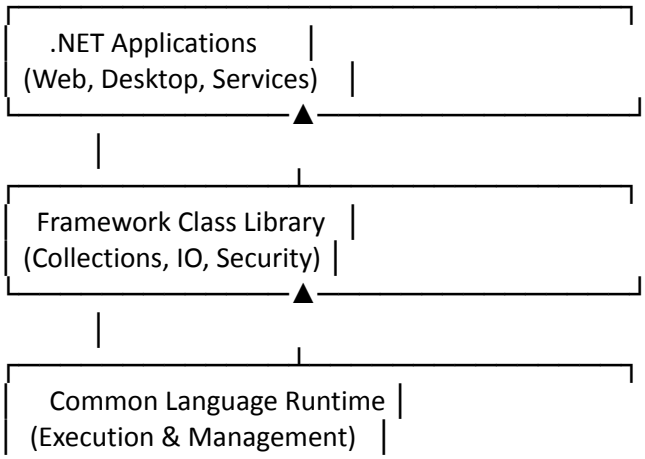
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.

IoT & Embedded Systems	Device control and telemetry
Data & Backend Services	Data processing, business services

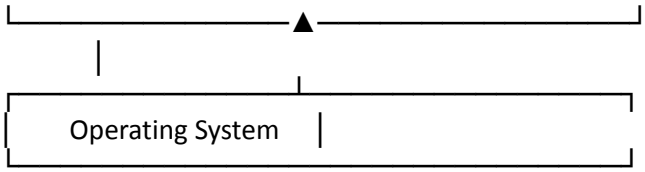
.NET framework architecture





Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.



Stages of C# Code compilation :

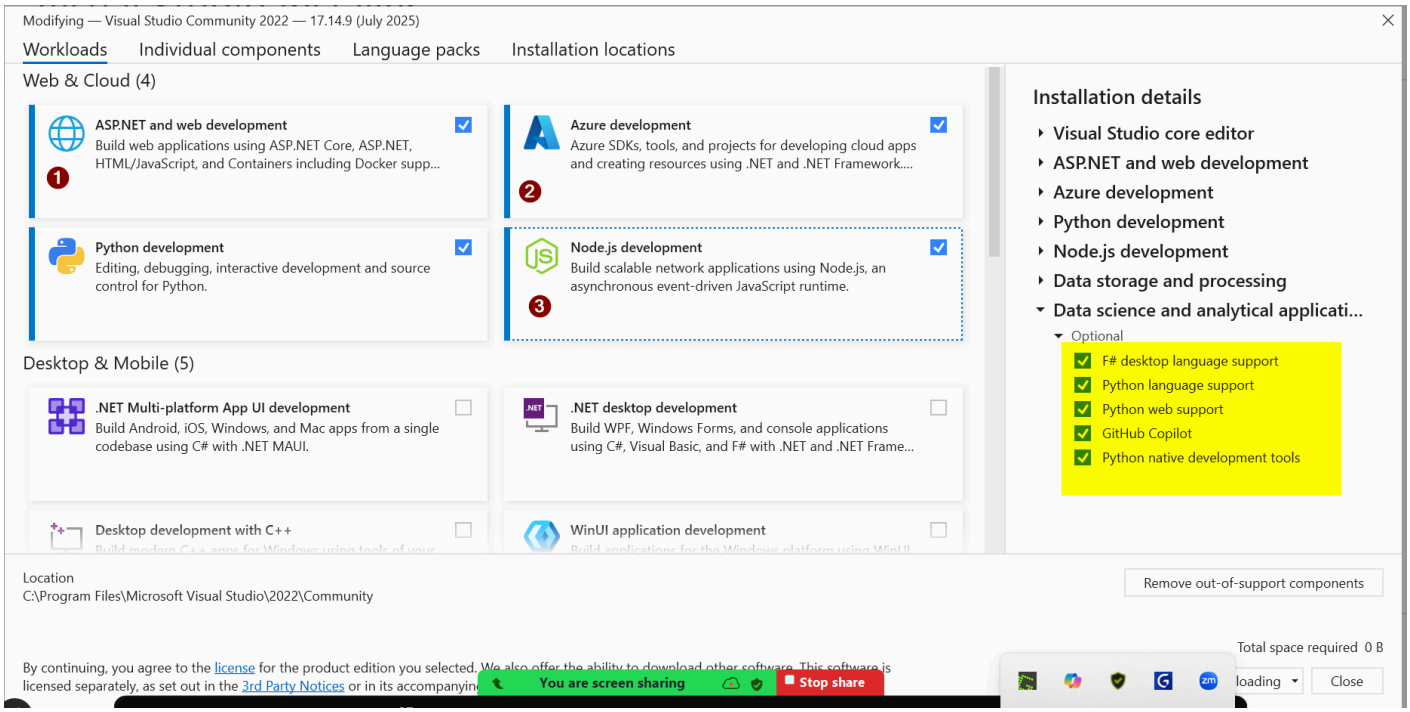
Step 1: C# Source Code (.cs)
Step 2: C# compiler (csc.exe) -> IL + meta data ⟶ Assembly (.exe /.dll) -> CLR loads assembly
Step 3: JIT compiler (just in time)--> Native machine code ⟶ Execution on CPU ⟶ output

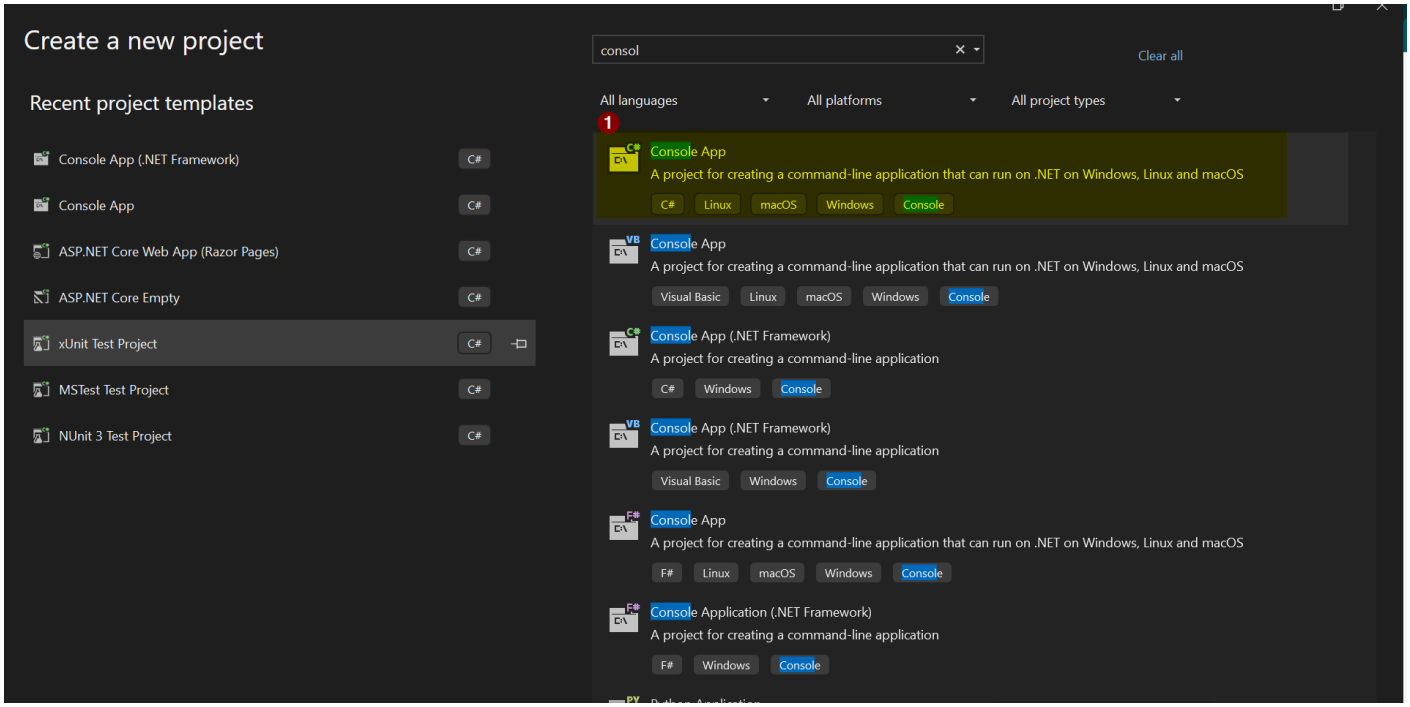
---

Aspect	.NET Framework	.NET Core	.NET 6+
Platform	Windows OS only	Cross-platform(linux/ mac)	Cross-platform
Performance	Moderate	High	Very high
Open Source	❌	✅	✅
Future Support	Legacy	Merged	Current & LTS
CLI support	Limited	Stronger	Full support
Deployment	machine wide install	Side by side	side by side
Cloud support	Weak	Strong	native a& optimized
Web framework	ASP.NET	ASP.NET Core	ASP.NET Core

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

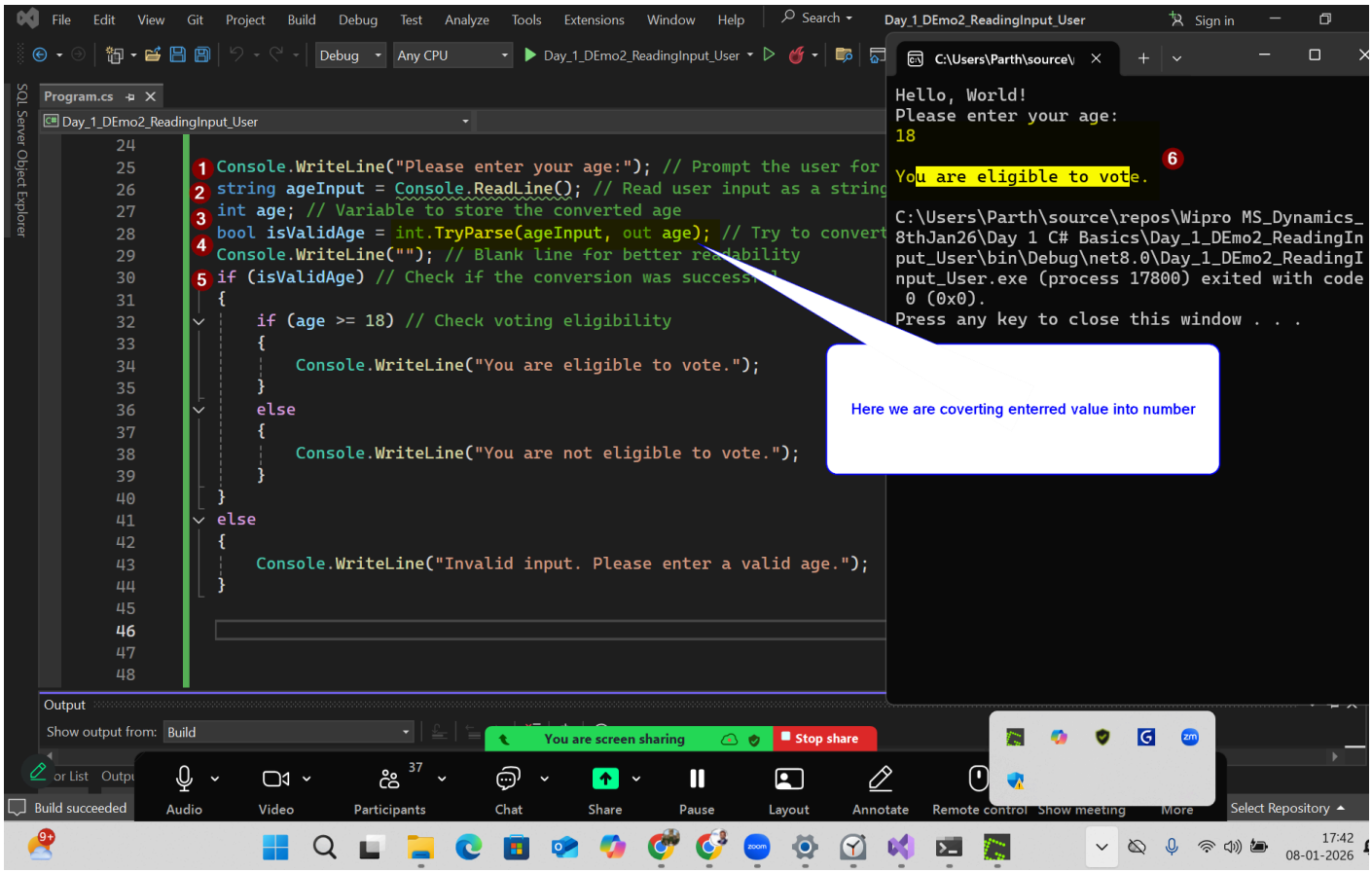
This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.





Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.



//Boxing is the process of converting a value type to an object type.
//Unboxing is the process of converting an object type back to a value type.

//Value type: Store value directly in memory (e.g., int, float, char).
//reference type: Store a reference (address) to the value in memory (e.g., string, arrays, class objects).

//Ex Storing money in wallet compared to storing money in bank account.
//in terms of access speed Value types are faster than reference types.
//IN terms of space efficiency Value types are more space-efficient than reference types.
//Value types are based on Stack memeory(fixed size) while reference types are based on Heap memory(unlimited).
//Boxing and unboxing can introduce performance overhead due to additional memory allocation and type conversion.
//ideally we should use less of boxing and unboxing in performance-critical applications.
//In C# Object type is the base type from which all other types derive. ex all types are devied from object type.
//types in C# :
//1. Value types: int, float, double, char, bool, struct, enum
//2. Reference types: string, arrays, class, interface, delegate(function pointer)

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.

Steps for pushing your code on github repo: via Visual studio :

Step 1: Create or opening Project in Visual Studio

Step 2: Sign in into Github from Visual Studio

Step 3: Initialize git repo using git menu -> create git repository

- **Choose GitHub as the remote**
- **Provide repository name**
- **Choose Public or Private**
- **Click Create and Push**

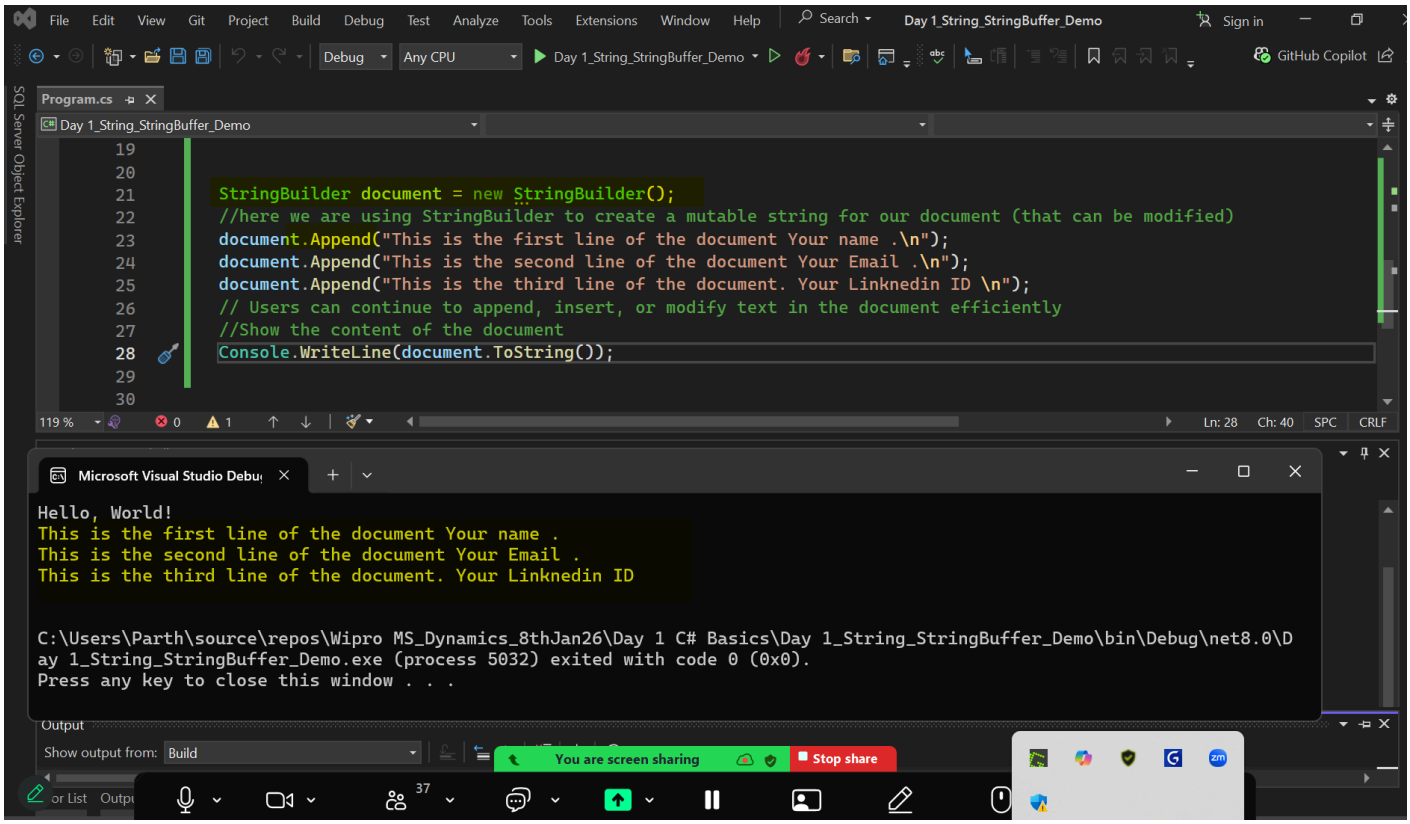
Step 4: review git changes and stage files finally commit changes so that code can be pushed to github

Step 5: verify changes on github

Step No.	Action	Command / Description
1	Open project in VS Code	File → Open Folder → Open project
2	Open terminal	Terminal → New Terminal
3	Initialize Git repository	`git init`
4	Check repository status	`git status`
5	Add files to staging	`git add .`
6	Commit changes	`git commit -m "Initial commit"`
7	Create GitHub repository	Create repo on GitHub (no README)
8	Add remote origin	`git remote add origin https://github.com/<username>/<repo>.git`
9	Verify remote	`git remote -v`
10	Rename branch to main	`git branch -M main`
11	Push code to GitHub	`git push -u origin main`
12	Authenticate (if prompted)	Login via browser
13	Verify on GitHub	Check files & commits

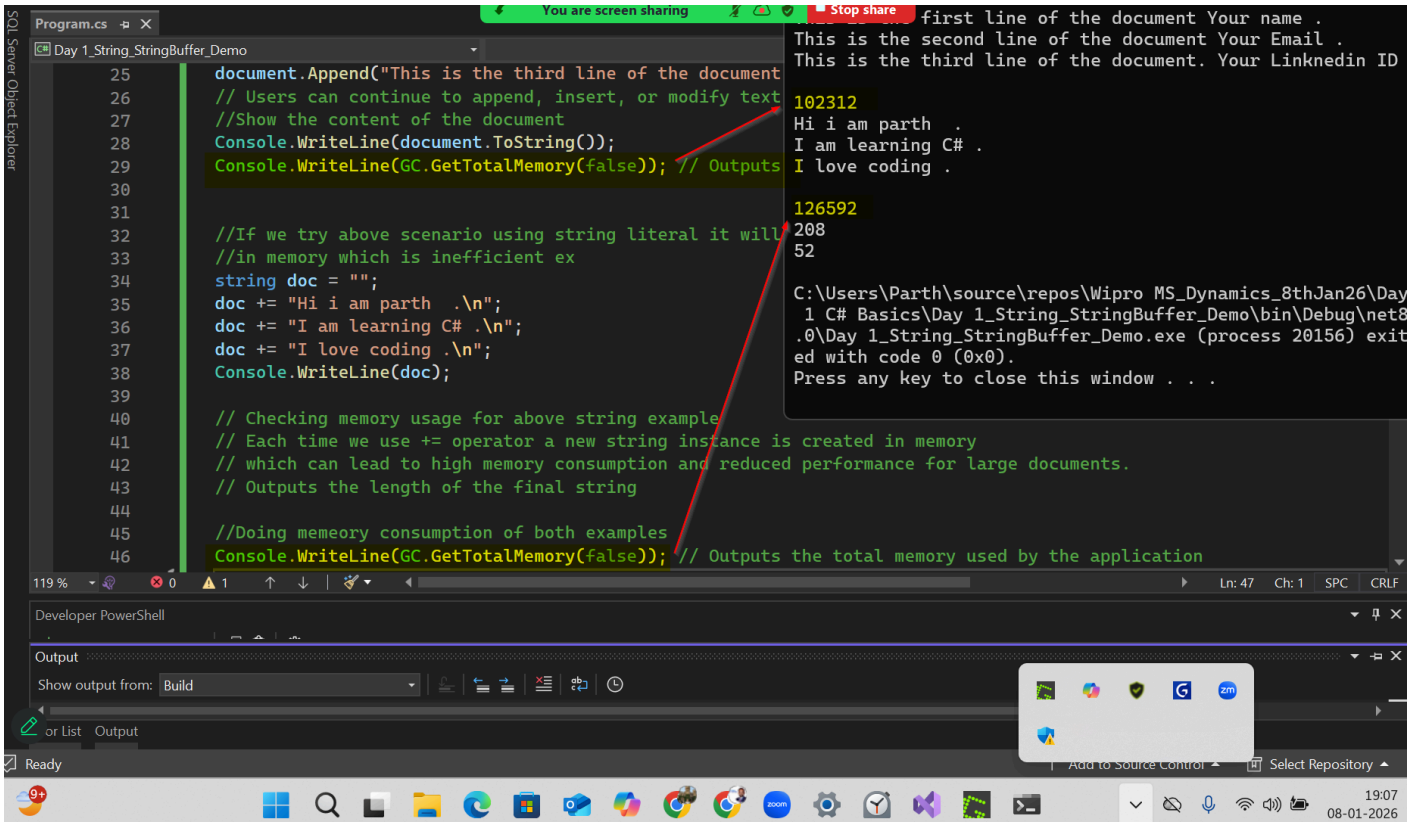
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.



Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.



Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

This file is meant for personal use by rushikeshsanjayjadhav2512@gmail.com only.
Sharing or publishing the contents in part or full is liable for legal action.