GAZİ UNIVERSITY

FACULTY OF ENGINEERING
DEPARTMENT OF INDUSTRIAL ENGINEERING

Lecturer : Dr Ercan Ezin

IE104-COMPUTER PROGRAMMING I

WEEK 3: INTRODUCTION PROGRAMMING AND C#

IMPORTANT ANNOUNCEMENT

- The course website is online now. Please check it out regularly.

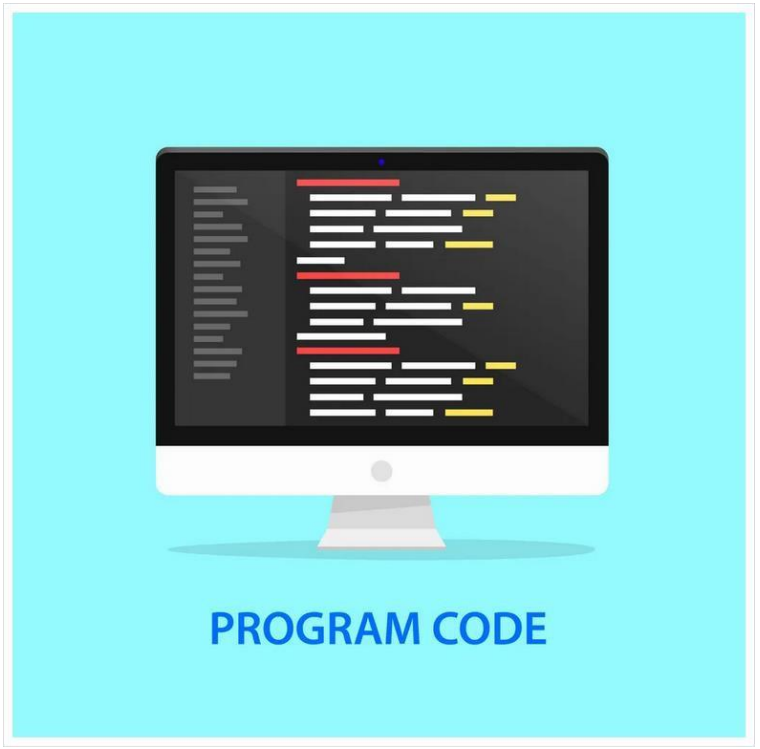
- https://gazi-end-muh.github.io

Programming Terms and Programming

- Program
- Programming
- IDE(Integrated Development Environment)
- Compiler
- Interpreter
- Execution
- Linkers
- Error Types
- Debug

What is a Program

- A set of expressions (commands, words, arithmetic operations, logical operations, etc.) created by using a computer to solve an existing problem is called a «program».



Programming

- A computer program is an organized set of instructions designed to solve complex problems requiring significant computational power.
- Programming is the process of creating these instructions through the use of a dedicated programming language.
- These languages range from high-level options like Java and C# to low-level systems like C, Assembly, or machine code.
- Source code is typically transformed into an executable via compilers and linkers, or executed line-by-line using an interpreter.

Programming

- Programmers usually start by writing the "Hello World" program.

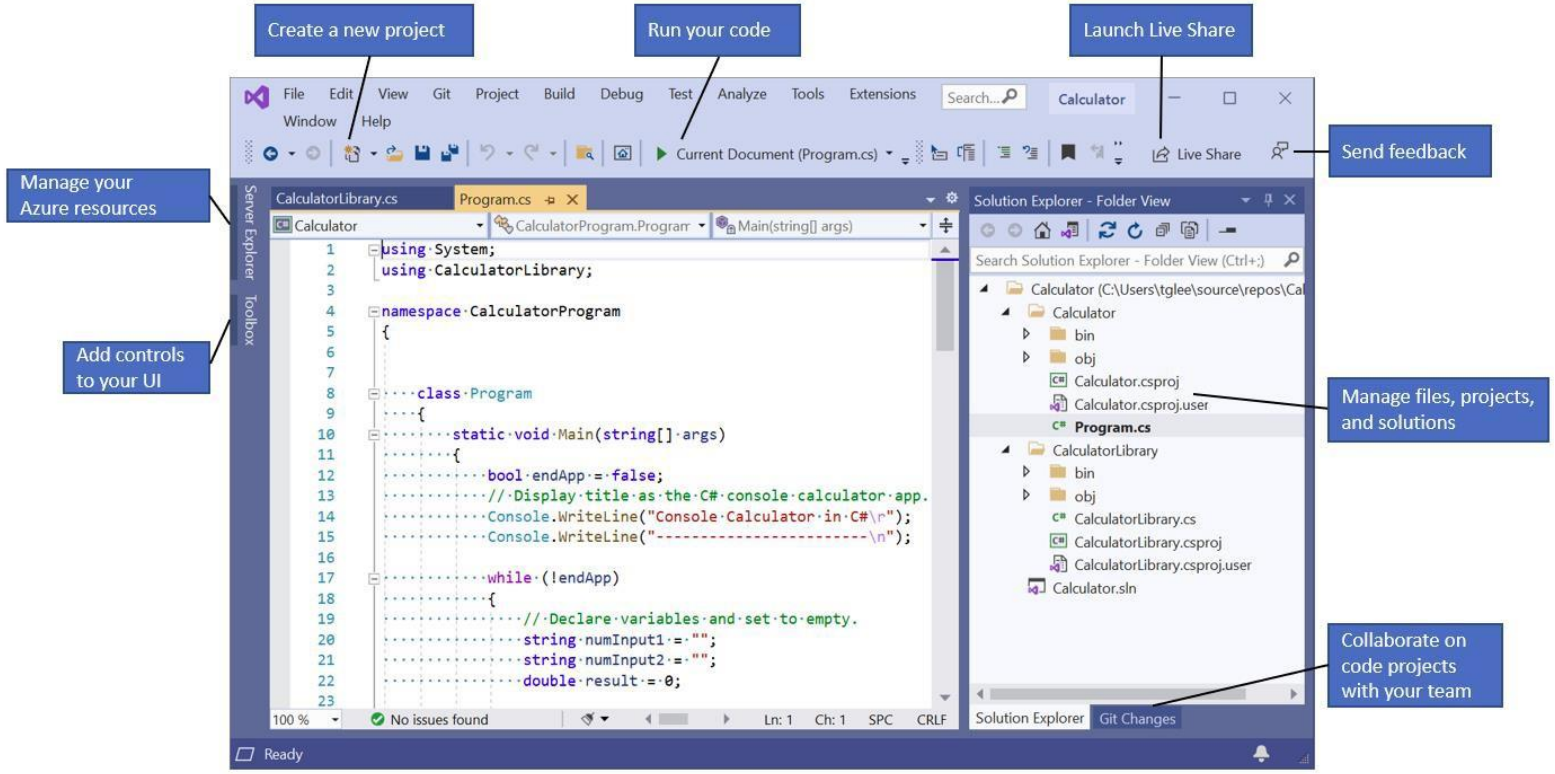
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp3
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");

        }
    }
}
```

IDE - Integrated Development Environment

- IDE is a type of software including all the tools that contribute to the efficient use of the development process. An IDE can organize the development process while aiming for the software to develop programs quickly and conveniently.



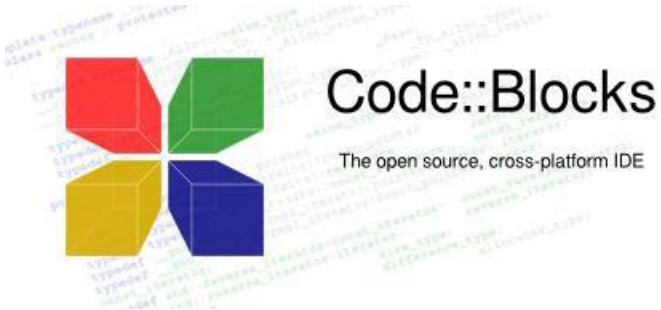
IDE - Integrated Development Environment

Essential Features of an IDE Syntax-Aware Script Editor:

- Employs language-specific color-coding to distinguish keywords, variables, and logic, enhancing code readability.
- Hierarchical File Management: Provides a structured, real-time navigation pane to organize project files and directories logically.
- Unified Development Toolkit: Consolidates a compiler, interpreter, and debugger into one interface for seamless code translation and error resolution.
- Automated Build Systems: Includes integrated tools to compile, link, and execute software automatically, streamlining repetitive development tasks.

IDE – Integrated Development Environment

- Most known Integrated Development Environments: Eclipse, Microsoft Visual Studio, Code::Blocks, Dev-C++, NetBeans…



Compiler

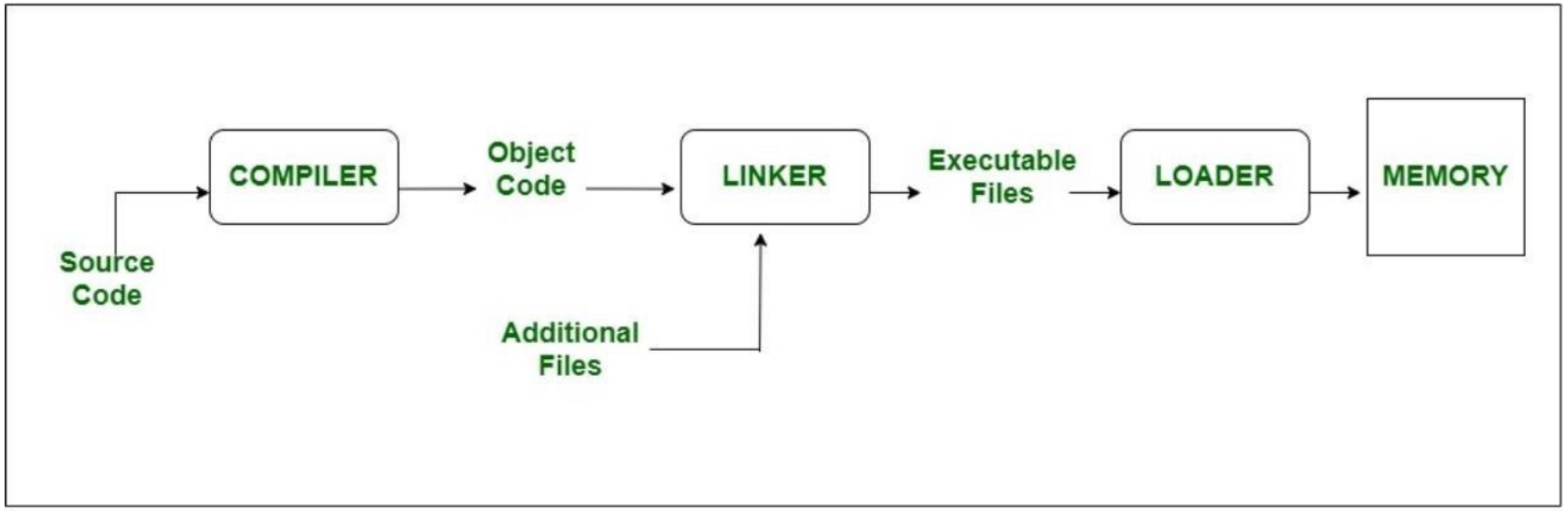
- The compiler scans the source code to detect syntax or logical errors, providing user feedback to facilitate debugging. Once verified as error-free, it translates the code into a machine-readable format for execution.

Interpreter

- The interpreterdirectly executes the source code taking it part by part.
- Interpreters do not make a standard executable code.
- Since the interpretation process is done step by step, the program is usually interrupted at the place where the first error is found.
- Unlike compilers; Unhandled lines of code are never passed over, and errors are not cared.
- Interpreters generally run slower than compilers because they instantly convert from source code to machine language. Also, code optimization is often not possible.

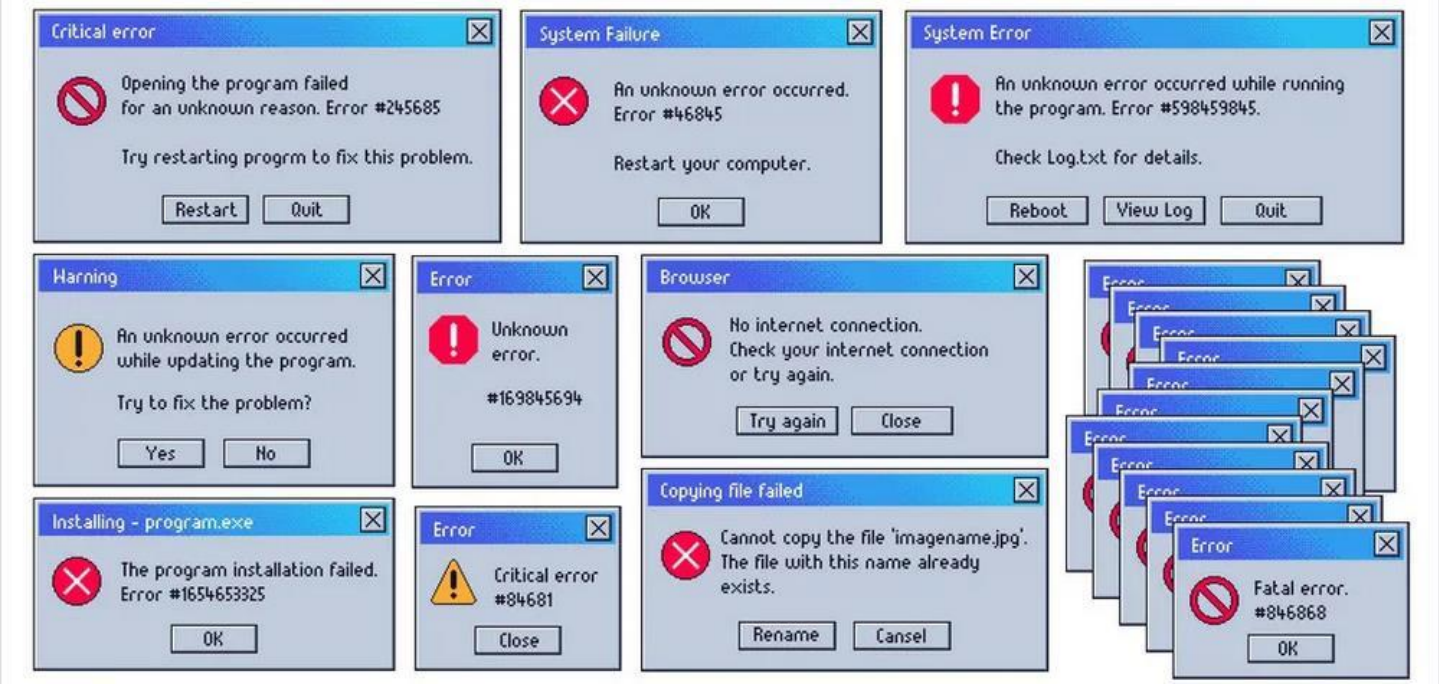
Linker and Execution

- Linker: Associates one or more files that are converted to object file by the compiler and convert them into a single executable file (For example, Windows.exe).
- Execution: The step of running the program.



Software Errors

- Software bugs are human errors occurring at any stage of development, including analysis, design, coding, testing, or maintenance. These mistakes are never intentional, and it is common for developers to overlook bugs while performing their own tests.



Software Errors

- It is practically not possible to build an error-free software program. However, by following certain standards and employing a defensive approach, one can reduce the number of errors and can be quicker to intervene when errors occur.
- In an application development, Errors are evaluated in 3 groups:
  1. Syntax Errors
  2. Run-time Errors
  3. Logical Errors(Bugs)

Syntax Errors

- These are the errors that can be encountered due to a number of expressions that are against the programming language rules in the written program.
- These are simple mistakes to fix.
- The line with the error is reported by the compiler.
- In today's IDEs, these problems are almost non-existent. Thanks to the advanced spelling of code editors, software developers can detect syntax errors without even needing to compile them.
- If a Syntax Error is received in compiling, it means the object file could not be produced.

Output:

```
HelloWorld.cs(17,16): error CS1525: Unexpected symbol `Console'
```

Runtime Errors

- These are the errors encountered during the execution of the program. When some inconsistent situations occur that the programmer does not take into account, the program is interrupted by the operating system.
- If the user has addressed these errors, the program can be terminated appropriately with the messages given by the programmer.

For example, followings are situations that most programmer faces daily:

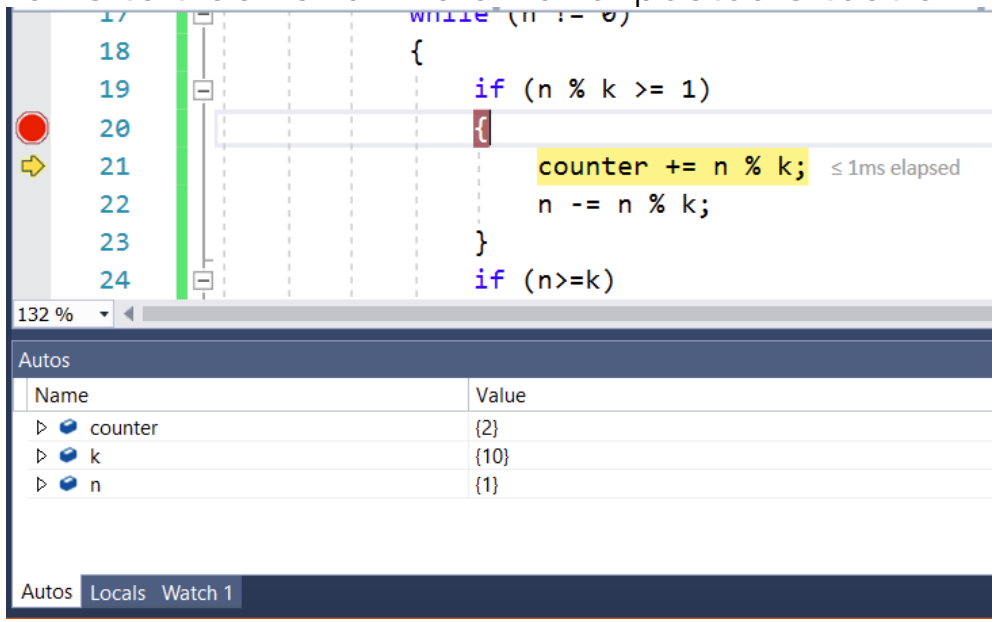
- trying to open a file that does not exist
- trying to allocate memory from a non-existent memory source
- trying to reach a non-existent file, etc.

Logical Errors (Bugs)

- It is a dangerous error that you may encounter. It is caused by some misunderstandings things in programming logic.
- For example; Incomplete or incorrect calculation of values that need to be calculated is a logical error. It may be necessary to return to the analysis phase to resolve this issue. Sometimes it is very difficult to find where this error originates from.
- Both free software and commercial software all contain logical errors that called bugs.
- Today, even the most active software companies admit that there are bugs in their software, and from time to time they produce an Update, Patch for their software to fix these bugs, or they release a new version of the software.

Debug

- This is the operation that is performed to fix logical errors or find bugs in the software. Generally, the written program is run step by step and in a controlled manner.
- It allows programmer to see the value of the relevant variables at each step of the program, and also allows to track and find an unexpected situation more easily.



What is C#

- C# Programming Language is an element of Microsoft's recently developed .NET platform.
- It was formed by blending old programming languages with a new language.
- It is also a derivative of C, C++, Java, Visual Basic languages.
- It is a 100% object-oriented language developed from the ground up for the powerful, simple, flexible, type safe .Net platform, derived from C#, C/C++ and Java languages.

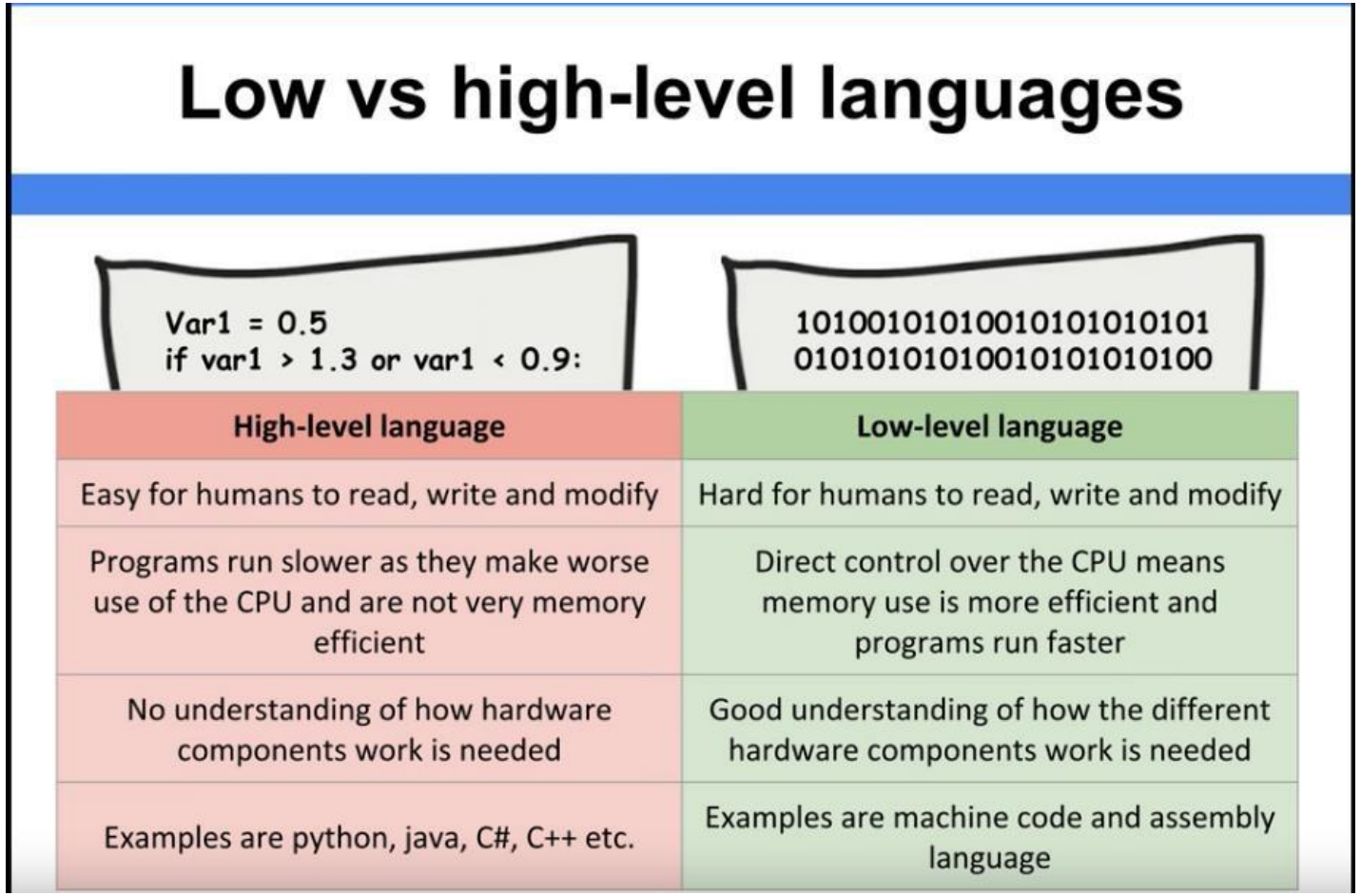
What is C#

- C# is known for its similarity to C/C++ and Java languages, which have been widely used by programmers. Although it seems to be a very similar language at first glance, it has many different features from these two languages.
- It takes the efficient features of C/C++ and Java and leaves out the potentially dangerous features of those languages.
- Unlike C/C++ language, C# is completely object oriented. Even basic data types such as “int”, “double” are defined as objects has their own object implementation that can be updated.
- As an example difference from Java language, a feature to manipulate Memory data called pointers can be used in C# language.

Why is C# more common and preferred?

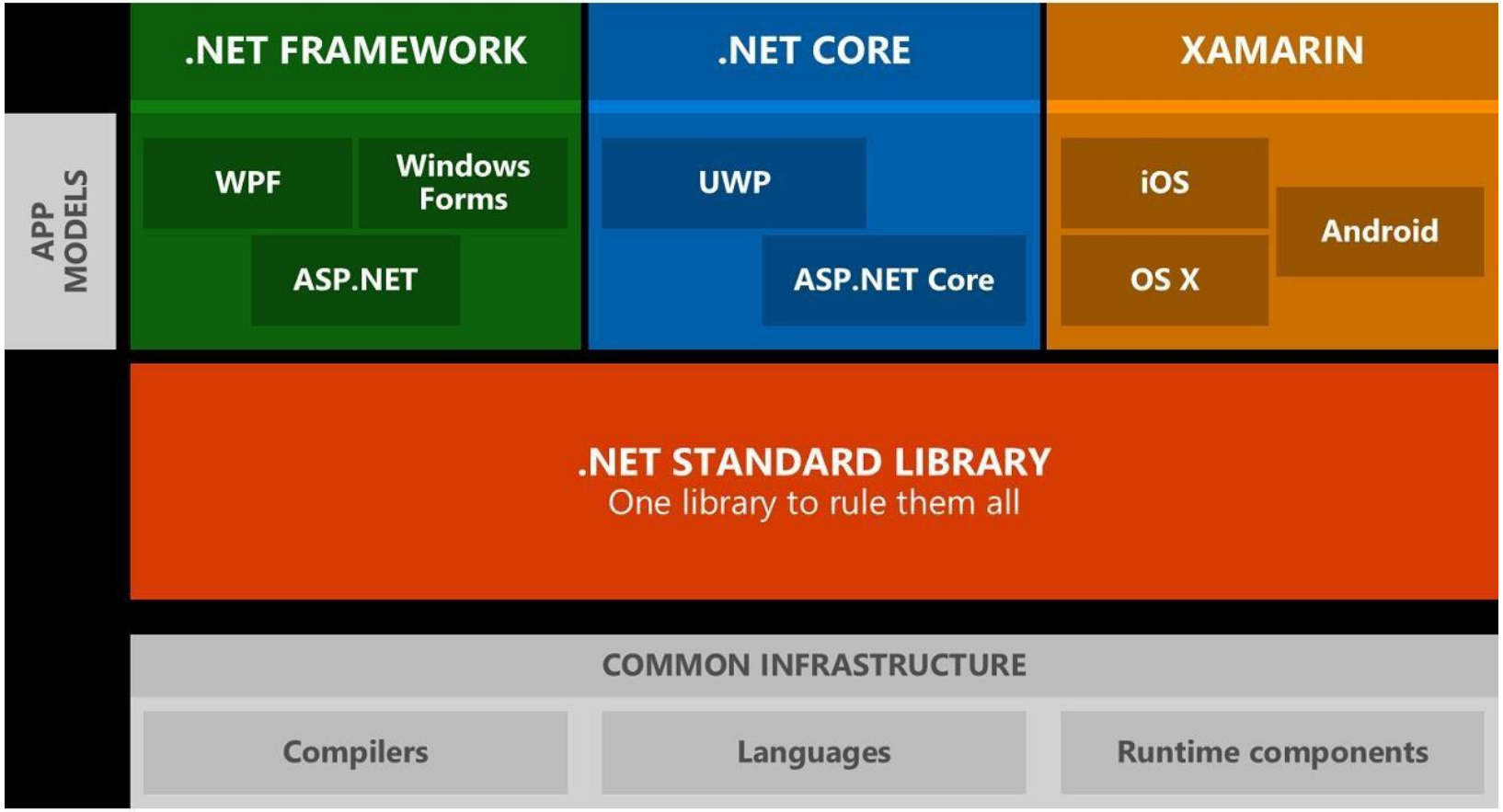
- Easy to learn.
- Provides full support for object-oriented programming.
- High efficiency.
- Balance between power and convenience.
- Offers XML support.
- Supports Windows Forms logic.
- Has adapted to the development of Internet Technologies as a modern language.

PL Classifying (Low-High Level)



High-level language	Low-level language
Easy for humans to read, write and modify	Hard for humans to read, write and modify
Programs run slower as they make worse use of the CPU and are not very memory efficient	Direct control over the CPU means memory use is more efficient and programs run faster
No understanding of how hardware components work is needed	Good understanding of how the different hardware components work is needed
Examples are python, java, C#, C++ etc.	Examples are machine code and assembly language

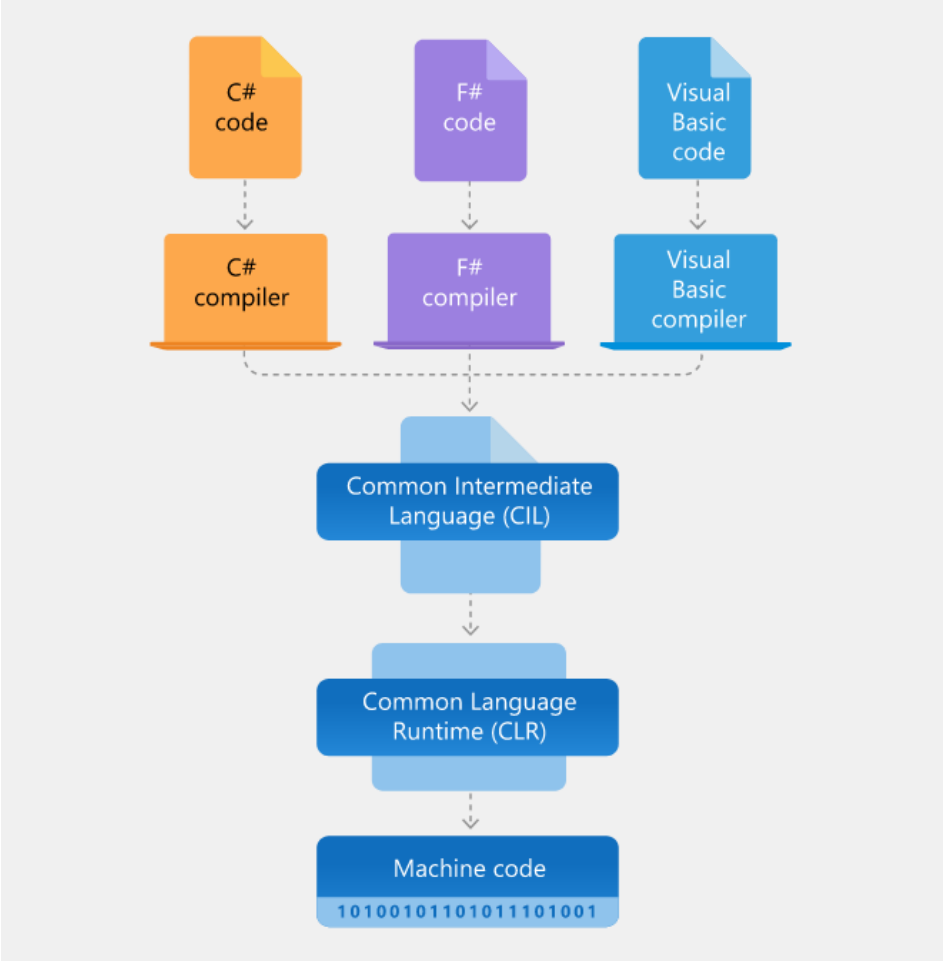
.Net Framework



.Net Framework

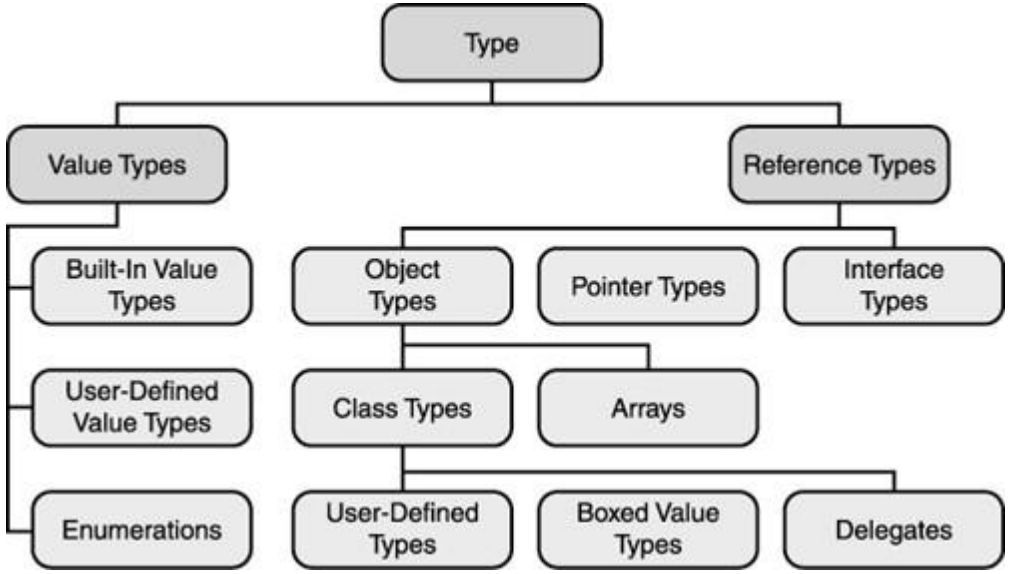
- .NET applications are written in the C#, F#, or Visual Basic programming language. Code is compiled into a language-agnostic Common Intermediate Language (CIL). Compiled code is stored in assemblies—files with a .dll or .exe file extension.

- When an app runs, the CLR takes the assembly and uses a just-in-time compiler (JIT) to turn it into machine code that can execute on the specific architecture of the computer it is running on.



CTS

- The Common Type System (CTS) is a standard for defining and using data types in the .NET framework.
- CTS defines a collection of data types, which are used and managed by the run time to facilitate cross-language integration.



Namespaces and .NET Class Library

- There are a number of basic types and classes that the .NET Framework offers programmers.
- In order to organize all these classes and types well, .NET uses the concept of namespace.
- Data types and classes in the .NET Framework class library in C# are used with the "using" keyword. In other languages, these namespaces are declared to the compiler in different ways.
- When developing a program, it is very important to put the related classes in the same namespace in terms of finding errors and intelligibility in the program.

Namespaces and .NET Class Library

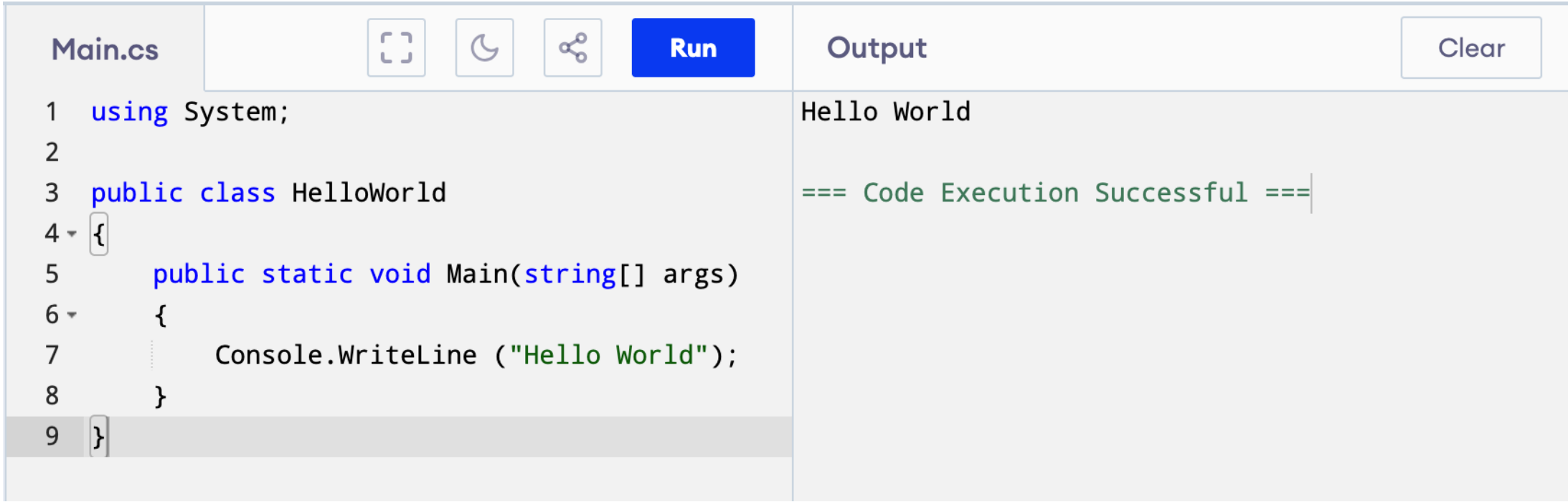
- The most frequently used class libraries in the .NET class library are:
  - ✓ System: Contains essential classes when working with .NET. Also all other class libraries are clustered within this namespace.
  - ✓ The basic class “Console” required for basic input and output operations in console-based applications, and the “Math” class, which includes many mathematical functions, are also included in the System namespace.
  - ✓ System is located at the top of the hierarchy.

Namespaces and .NET Class Library

- System.Data: The class library that comes ready for all database operations is accessed with this namespace.
  - The namespace "System.Data.SqlClient" is available for operations with SQL in this class library.
- System.Xml: Contains the necessary limits for working with XML, one of the most used technologies for formatting data and sharing data over the internet.
- System.Net: It is the namespace that contains the network components required for developing distributed applications. It is used for HTTP and network protocols.
- System.IO: It contains the necessary operations to work with (read/write) files.
- System.Windows.Forms: It is the namespace that contains the visual controls used in Windows-based applications.

ONLINE COMPILERS

- https://www.programiz.com/csharp-programming/online-compiler/
- https://onecompiler.com/csharp



Compiling and Running a C# program

HelloWorld.cs +

```
1  // C# program to print basic output to the console
2  using System;
3
4  // A logical container for your code (Namespace)
5  namespace ProgrammingLab
6  {
7      // The main class for the application
8      class Program
9      {
10         // The Entry Point where the program starts execution
11         static void Main(string[] args)
12         {
13             // --- Basic Output (Topic 3) ---
14             Console.WriteLine("Welcome to Intro to Programming 1!");
15             Console.WriteLine("This is a standard C# program structure.");
16
17             // Add an empty line for readability
18             Console.WriteLine();
19
20             // --- Preventing the window from closing immediately ---
21             Console.WriteLine("Press any key to exit...");
22             Console.ReadKey();
23         }
24     }
25 }
```

- First, open one of the online compiler given.
- Write the code in the text editor and hit the run button.

Important Issues

```
class first_program
    {
    static void Main()
        {
            System.Console.WriteLine("Hello C#");
         }
    }
```

- **Main** function is the location where your program always starts and must exists.
- C# is a 100% object-oriented language. Everything is an object. In C and C++, the program execution starts from the main function, but the main function has never been inside a class.
- Since everything is represented by classes in C#, the main function must also be a part of a class.

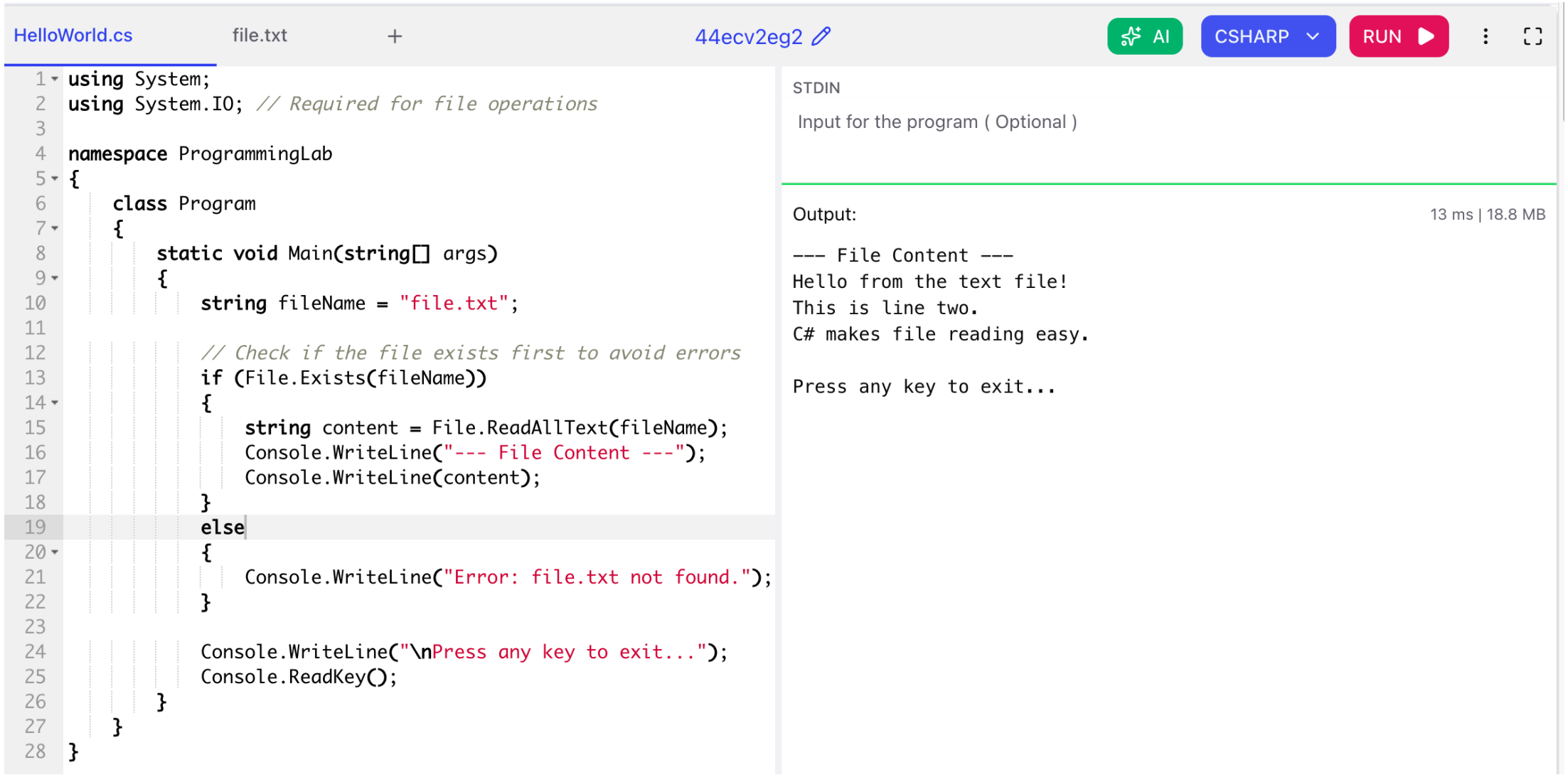
Tips and Tricks!

- All C# programs must contain at least one class. Programs that are not inside the class declaration will not compile.
- Main() function is the starting point of the program.
- In C# all lines in the source code are terminated with "Semicolon-;". (Except in some cases.)
- Classes and functions are written inside opening and closing curly braces { } which is also called scope of a program.
- Many concepts in C# are built on objects called classes. Each class has several elements that executes some instructions. These executive elements are called methods or functions.
- The ReadLine() method is used like WriteLine(), but nothing is written inside ReadLine method's parentheses.

What is an Online C# Compiler?

- An online compiler is a web-based environment that allows you to write, edit, and execute code directly in your browser without installing any software locally.
- How it Works (The Technical Side)
- Most online compilers, like Programiz, use a **client-server architecture**:
- **The Interface:** The website provides a code editor (often based on the **Monaco Editor**, which powers VS Code).
- **The Server:** When you click "Run," the code is sent to a remote server.
- **Compilation:** A backend compiler (like **Roslyn** for C#) translates your code into **Intermediate Language (IL)** or machine-readable instructions.
- **Execution & Output:** The server executes the program within a secure, isolated "sandbox" to protect the system. The final output is then sent back and displayed in your browser console.

User Interface of an online compieler




```
using System;
using System.IO; // Required for file operations

namespace ProgrammingLab
{
    class Program
    {
        static void Main(string[] args)
        {
            string fileName = "file.txt";

            // Check if the file exists first to avoid errors
            if (File.Exists(fileName))
            {
                string content = File.ReadAllText(fileName);
                Console.WriteLine("--- File Content ---");
                Console.WriteLine(content);
            }
            else
            {
                Console.WriteLine("Error: file.txt not found.");
            }

            Console.WriteLine("\nPress any key to exit...");
            Console.ReadKey();
        }
    }
}
```

https://onecompiler.com/csharp/

Advantages of using online compilers

- **Zero Installation:** You can start coding immediately without managing heavy IDE installations like Visual Studio.
- **Universal Accessibility:** It works on any operating system (Windows, Mac, or Linux) and even on Chromebooks or tablets.
- **AI-Assisted Learning:** Modern platforms like Programiz and CodeChef now include **AI mentors** that explain syntax errors in plain English, which is highly beneficial for beginners.
- **Simplified Interface:** By hiding complex project files and configuration settings, it allows users to focus purely on the logic of Algorithms and Flow Charts.

Disadvantages of using online compilers

- **Internet Dependency:** A stable connection is required. If you lose connectivity, you cannot compile or run your code.
- **Execution Limits:** To prevent abuse, online compilers often have **time limits**. If a user accidentally writes an "infinite loop" (a common mistake), the server will kill the process abruptly.
- **Limited Low-Level Access:** While great for basic logic, online tools often restrict access to the file system (I/O) or complex external libraries.

Assignment

- Work on examples given on course website

THE END

CHECK OUT COURSE WEBSITE: HTTPS://GAZI-END-MUH.GITHUB.IO

GOT ANY QUESTIONS LATER?

SEND ME AN EMAIL: DR.ERCAN.EZIN@GMAIL.COM