**1. OVERVIEW OF C#**

C# (pronounced *C Sharp*) is an **object-oriented programming language** developed by Microsoft, mainly used for building **.NET applications** — desktop, web, and cloud.

**2. CLR (COMMON LANGUAGE RUNTIME)**

* CLR is the **heart of the .NET Framework**.
* It provides a runtime environment to execute programs written in .NET languages (like C#, VB.NET, F#).
* It handles:
  + **Memory management**
  + **Exception handling**
  + **Security**
  + **Garbage collection**
  + **Just-In-Time (JIT) compilation**

**Example flow:**  
C# Code → Compiled into **MSIL (Microsoft Intermediate Language)** → Executed by **CLR using JIT compiler**

**3. JIT (JUST-IN-TIME COMPILER)**

* Converts **Intermediate Language (IL)** into **machine code** just before execution.
* Makes .NET applications **platform-independent** and efficient.
* Types of JIT:
  + **Pre-JIT:** Compiles entire code at once.
  + **Econo-JIT:** Compiles only methods called at runtime.
  + **Normal JIT:** Compiles code on demand and stores it for reuse.

**4. GARBAGE COLLECTOR (GC)**

* Automatically **frees up memory** that’s no longer in use.
* No need for manual memory management (like in C/C++).
* GC works in **three generations (0, 1, 2)** to optimize performance.
* You can also call it manually (though rarely needed):
* GC.Collect();

**5. MANAGED CODE**

* Code that runs **under CLR supervision** is called *managed code*.
* CLR manages memory, security, and exceptions for it.
* Code outside CLR (like native C++) is *unmanaged code*.

**Example:**

Console.WriteLine("This is managed code under CLR!");

**6. VARIABLES & DATA TYPES**

**Variables:** Named storage for data.  
**Syntax:**

datatype variableName = value;

**Example:**

int age = 25;

string name = "Udaya";

float salary = 25000.5f;

**Common Data Types:**

|  |  |  |
| --- | --- | --- |
| **Type** | **Size** | **Example** |
| int | 4 bytes | 10 |
| float | 4 bytes | 12.5f |
| double | 8 bytes | 45.67 |
| char | 2 bytes | 'A' |
| string | Variable | "Hello" |
| bool | 1 byte | true/false |

**7. OPERATORS**

Operators are used to perform operations on variables.

**Types:**

* Arithmetic: + - \* / %
* Relational: == != > < >= <=
* Logical: && || !
* Assignment: = += -= \*= /=
* Increment/Decrement: ++ --

**Example:**

int a = 10, b = 20;

Console.WriteLine(a + b); // 30

**8. CONDITIONAL STATEMENTS**

Used to control flow based on conditions.

**if-else**

int num = 5;

if(num > 0)

Console.WriteLine("Positive");

else

Console.WriteLine("Negative");

**switch**

int day = 2;

switch(day)

{

case 1: Console.WriteLine("Monday"); break;

case 2: Console.WriteLine("Tuesday"); break;

default: Console.WriteLine("Other day"); break;

}

**9. LOOPS**

Used to execute code repeatedly.

**for**

for(int i = 1; i <= 5; i++)

Console.WriteLine(i);

**while**

int i = 1;

while(i <= 5)

{

Console.WriteLine(i);

i++;

}

**do-while**

int i = 1;

do

{

Console.WriteLine(i);

i++;

} while(i <= 5);

**Summary**

Learned: CLR, JIT, GC, Managed Code, Variables, Data Types, Operators, If-Else, Loops  
Practiced basic console programs.

**Snapshots (Day -1)**

**A screenshot of a computer

AI-generated content may be incorrect.Program**: Hello World & Managed Code

**A screenshot of a computer

AI-generated content may be incorrect.Program**: Variables and Data Types

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.**Program:** Operators

A screenshot of a computer

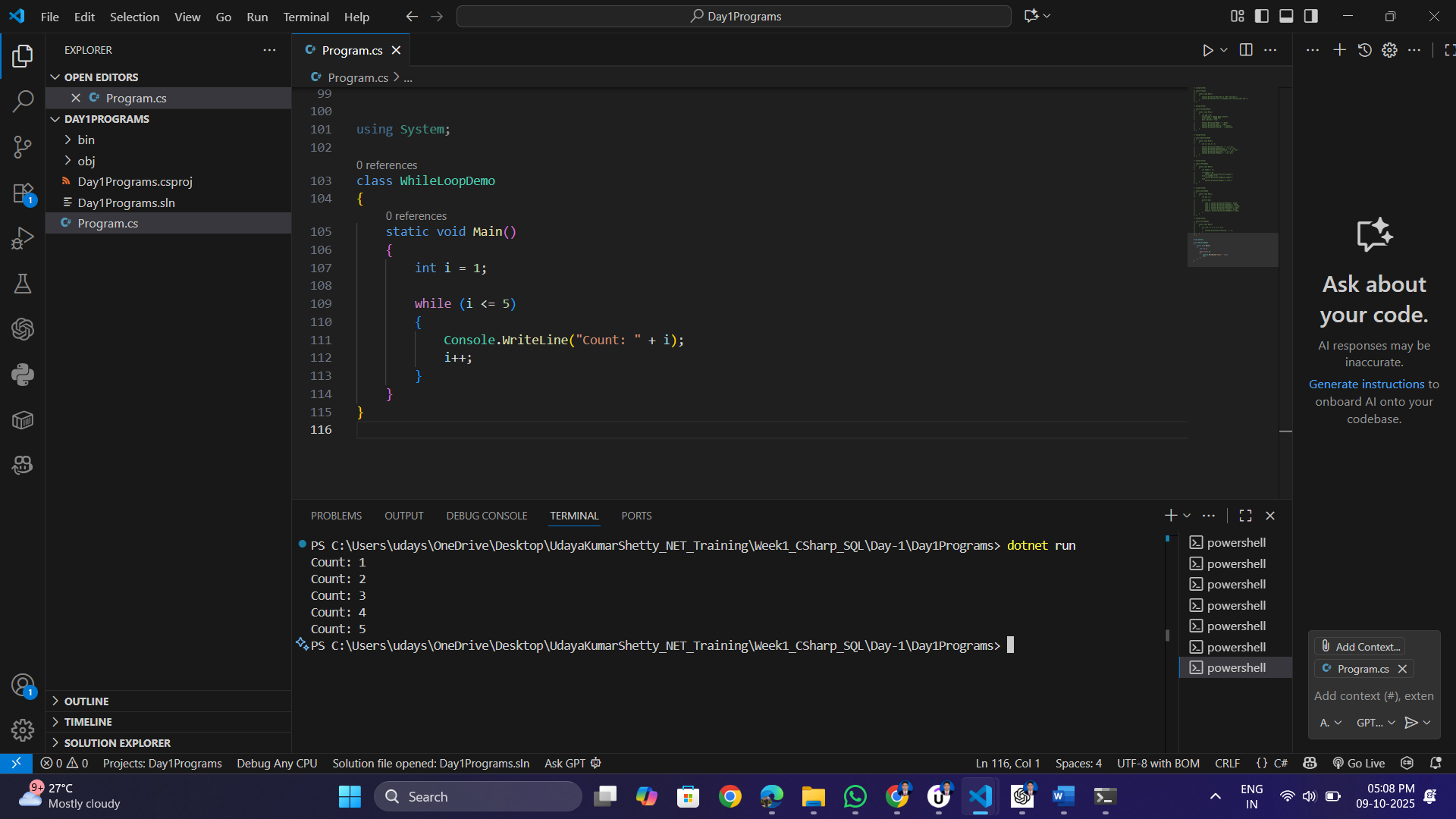
AI-generated content may be incorrect.**Program:** If–Else Example

A screenshot of a computer

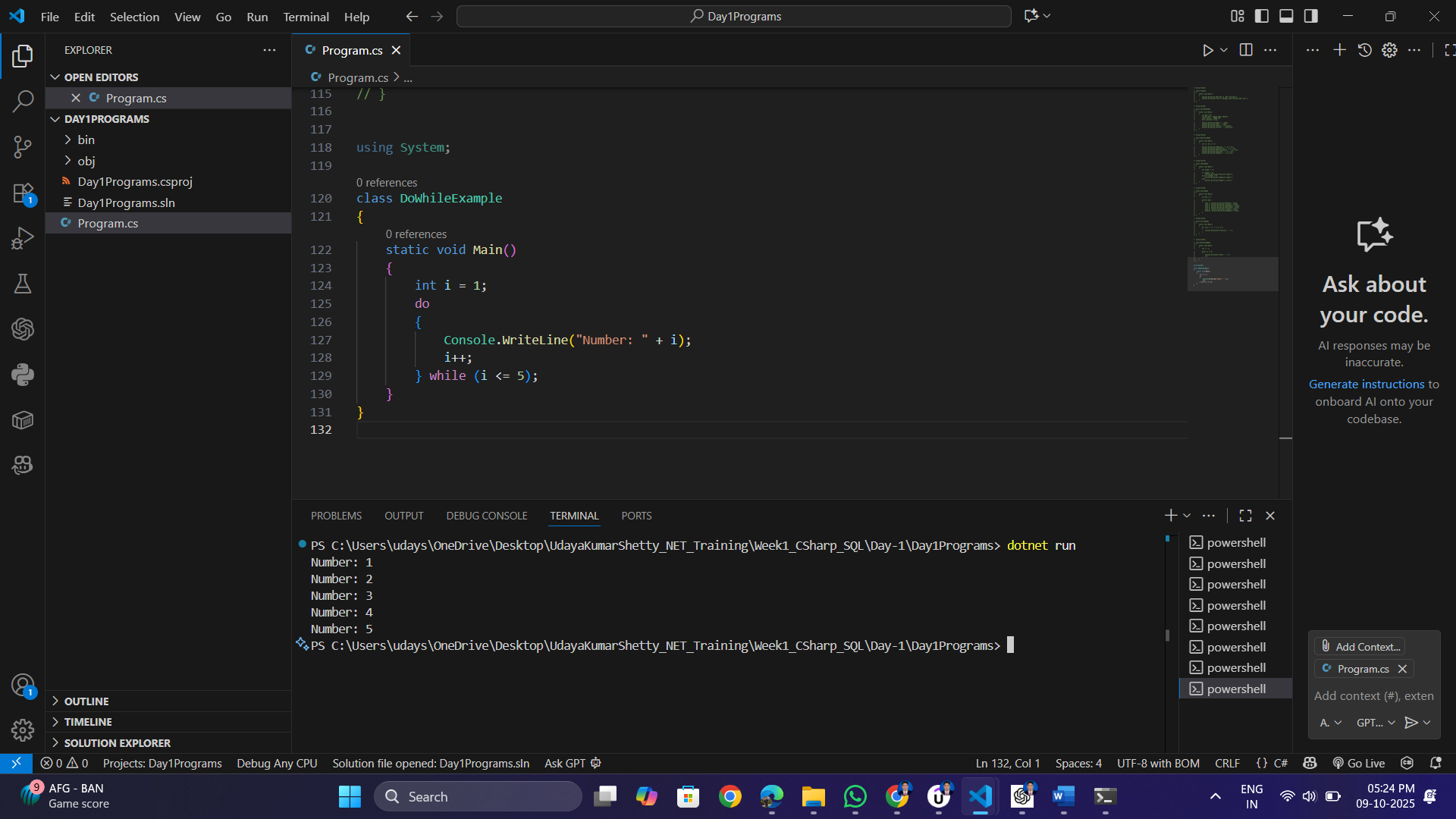
AI-generated content may be incorrect.**Program**: Switch Case Example

A screenshot of a computer

AI-generated content may be incorrect.**Program:** For Loop

**Program:** While Loop

**Program:** Do–While Loop

****