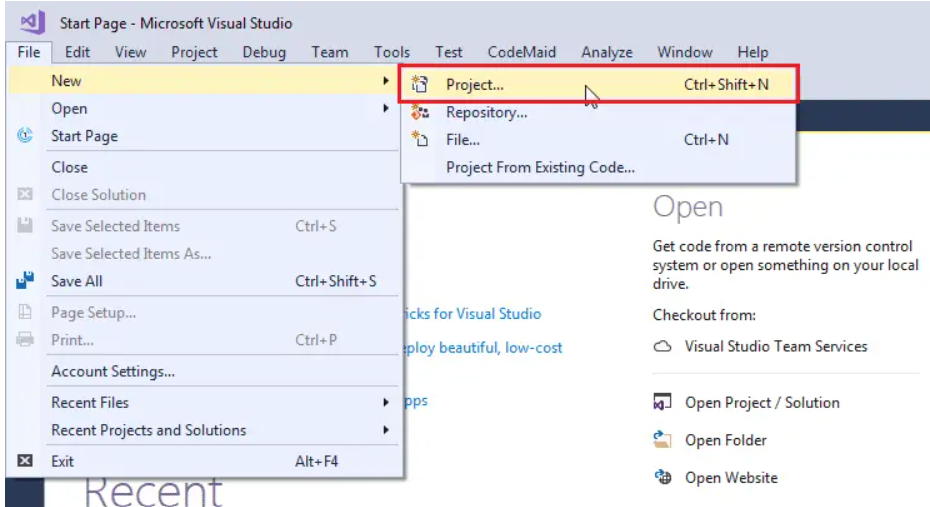
Lab # 4: C# Language Fundamentals:

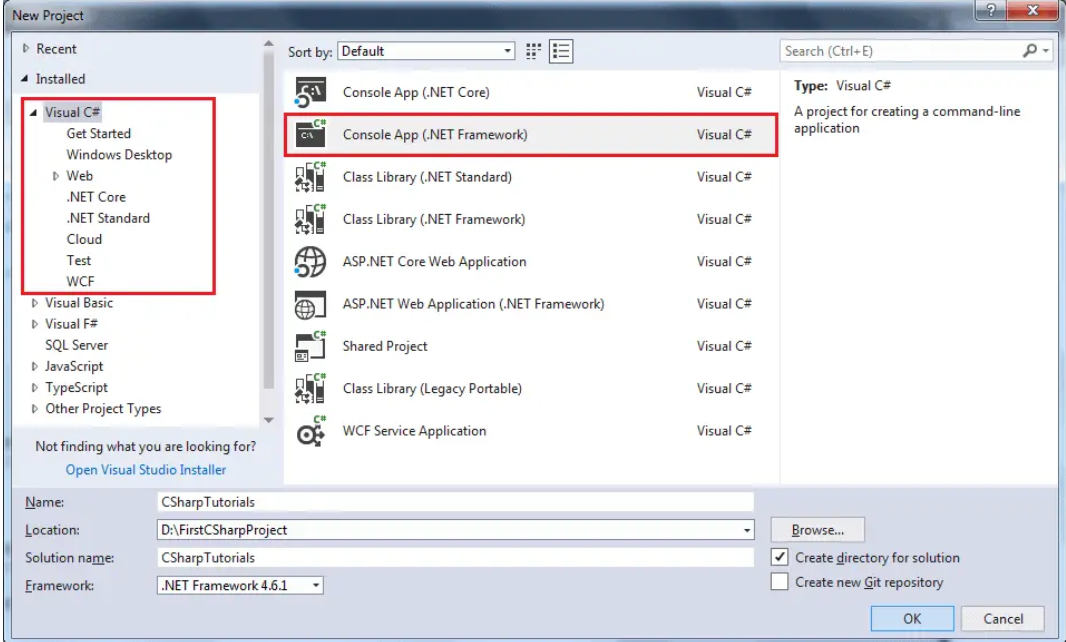
# Creating a Console Application:

C# can be used in a window-based, web-based, or console application. To start with, we will create a console application to work with C#.

Open Visual Studio installed on your local machine. **Click** **on File -> New Project**... from the top menu, asshown below.

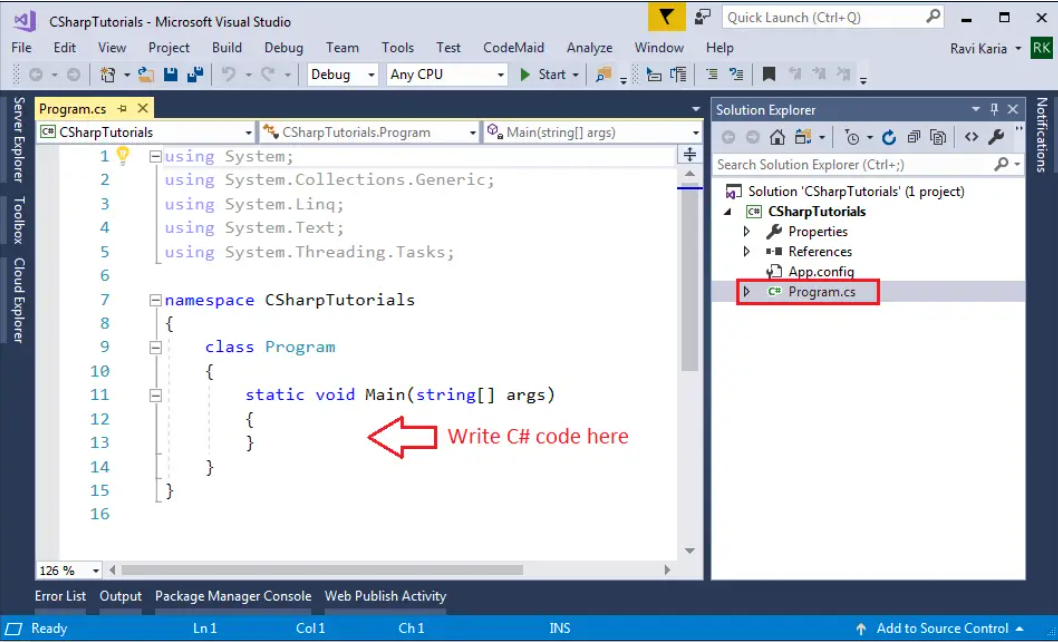


From the **New Project** popup, shown below, select Visual C# in the left side panel and select the Console App in the right-side panel.



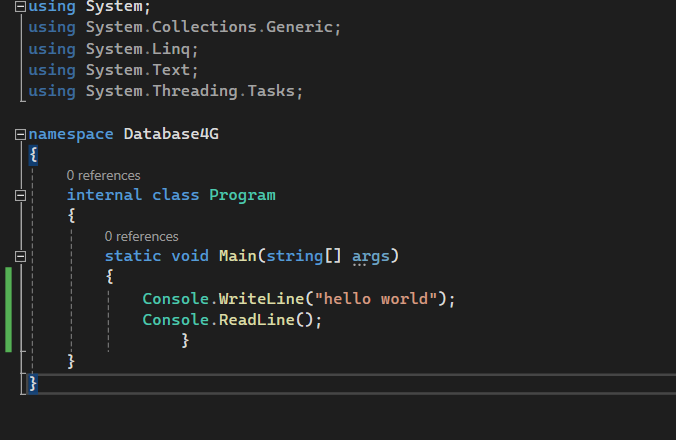
In the name section, give any appropriate project name, a location where you want to create all the project files, and the name of the project solution.

Click OK to create the console project. **Program.cs** will be created as default a C# file in Visual Studio where you can write your C# code in Program class, as shown below. (The .cs is a file extension for C# file.)



## Output

To output values or print text in C#, you can use the WriteLine() or Write()  method:

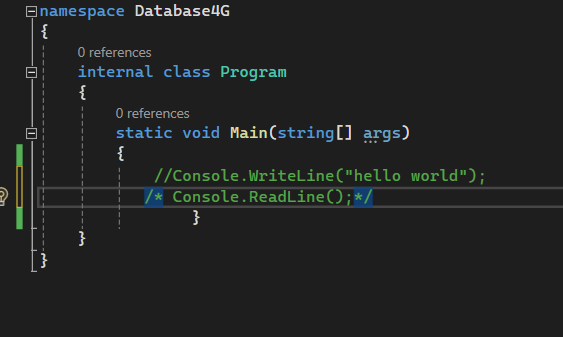


## Single-line Comments

Single-line comments start with two forward slashes (//).

## C# Multi-line Comments

Multi-line comments start with /\* and ends with \*/.



## Data Type:

In C#, there are different **types** of variables (defined with different keywords), for example:

int - stores integers (whole numbers), without decimals, such as 123 or -123

double - stores floating point numbers, with decimals, such as 19.99 or -19.99

char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes

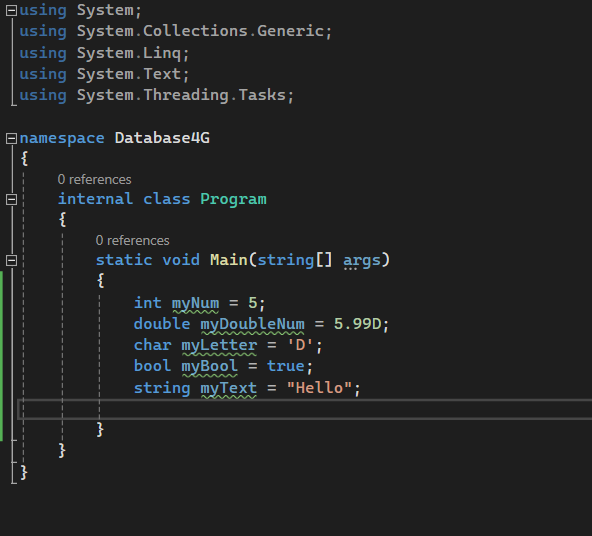
string - stores text, such as "Hello World". String values are surrounded by double quotes

bool - stores values with two states: true or false

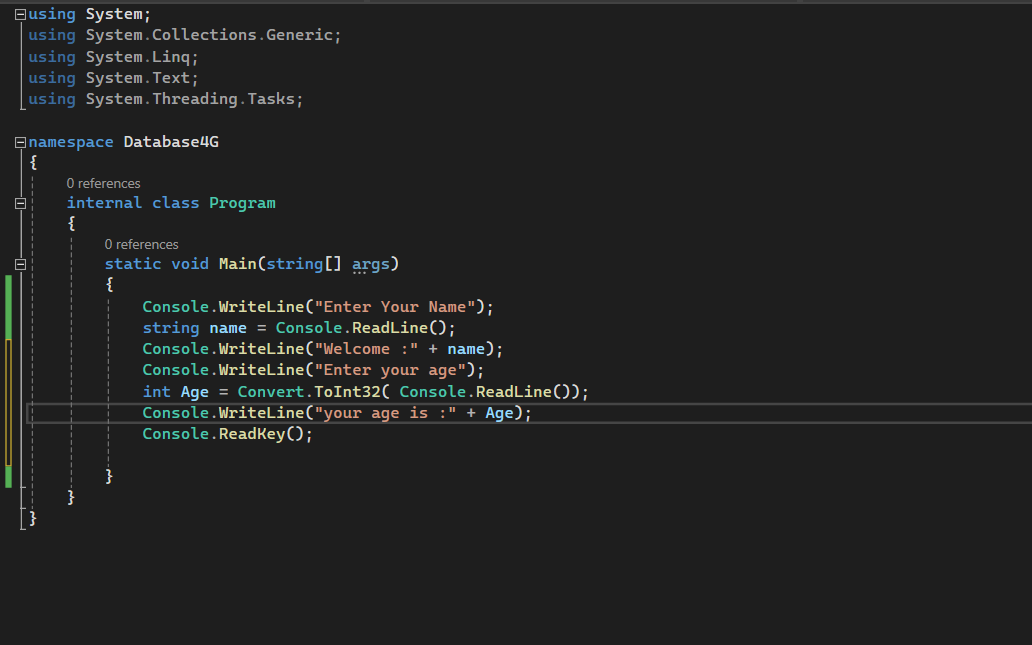
## Variable:

To create a variable, you must specify the type and assign it a value:

Type name\_of\_variable = data;



## User input:



## Operators:

* C# supports various operators such as arithmetic operators (**+**, **-**, **\***, **/,++**), assignment operators (**=**, **+=**, **-=**), comparison operators (**==**, **!=**, **<**, **>**), logical operators (**&&**, **||**, **!**), etc.
* Operators perform specific operations on operands.

*int a = 10;*

*int b = 5;*

*int result = a + b; // Addition*

*result = a \* b; // Multiplication*

*result = a > b ? a : b; // Conditional Operator*

## Control Structures:

* Control structures determine the flow of execution in a program.
* Common control structures include if statements, switch statements, loops (for, while, do-while), and jump statements (**break**, **continue**, **return**).
* These structures help in making decisions and looping over code blocks.

**If Statement**

if (condition)

{

// block of code to be executed if the condition is True

}

**Else Statement**

if (condition)

{

// block of code to be executed if the condition is True

}

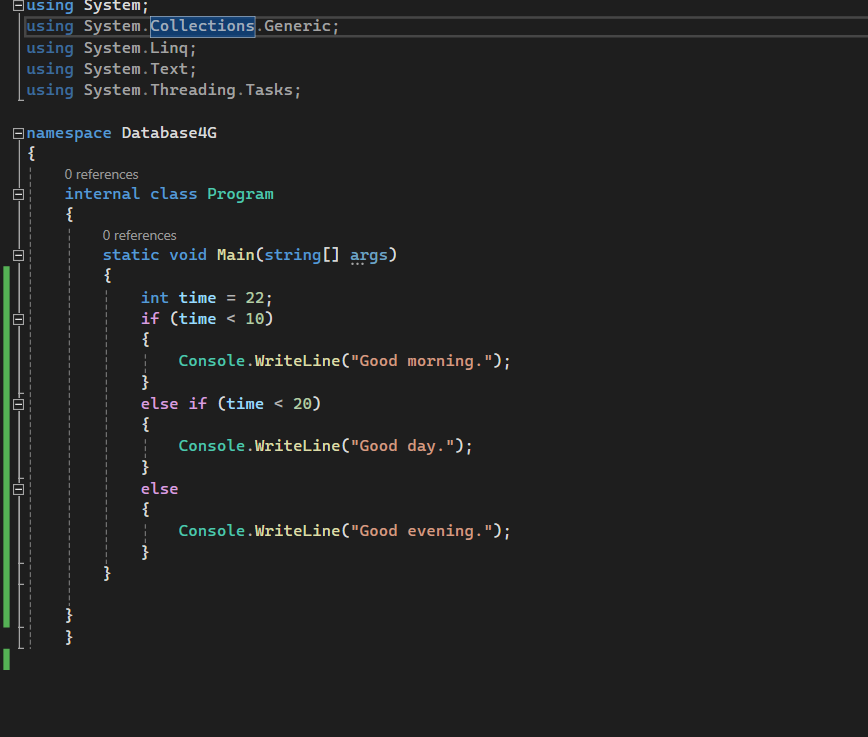
else

{

// block of code to be executed if the condition is False

}

**Else If Statement:**

****

**Switch Statement:**

switch(*expression*)

{

case x:

*// code block*

break;

case y:

*// code block*

break;

default:

*// code block*

break;

}

**Loops**

* **For Loop:**

for (statement 1; statement 2; statement 3)

{

// code block to be executed

}

* **While Loop:**

while (condition)

{

// code block to be executed

}

* **Do while Loop:**

do

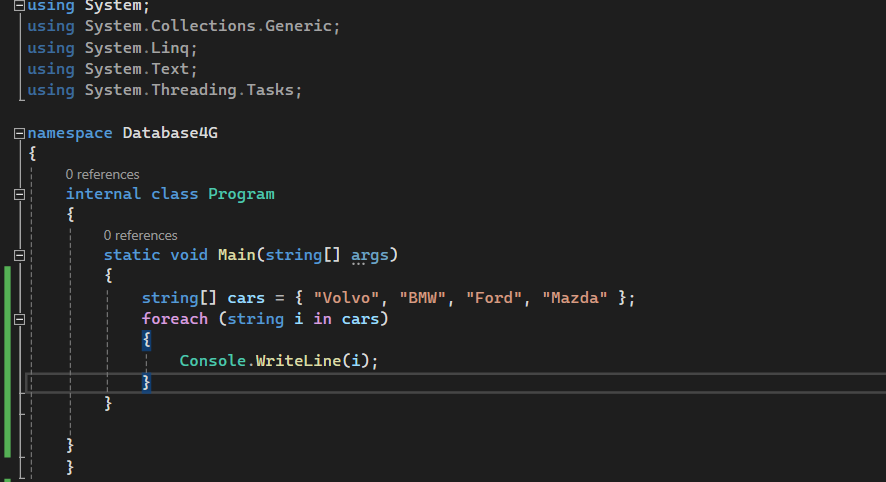
{

// code block to be executed

}

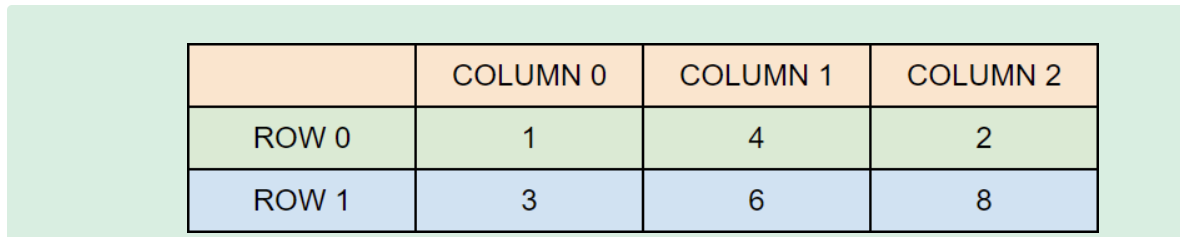
while (condition);

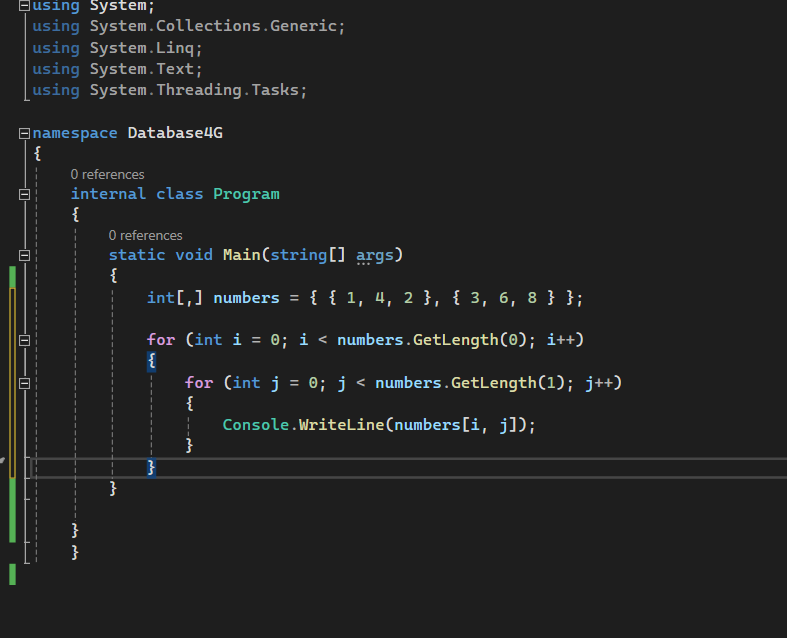
* **Foreach Loop:**

****

## Multidimensional Arrays

int[,] numbers = { {1, 4, 2}, {3, 6, 8} };



****

## OOP in C#:

using System;

public class Person

{

public string Name { get; set; }

public int Age { get; set; }

// Constructor that takes name and age as parameters

public Person(string name, int age)

{

Name = name;

Age = age;

}

public void PrintInfo()

{

Console.WriteLine("Name: " +Name + "Age:" + Age);

}

public void SetInfo(string name, int age)

{

Name = name;

Age = age;

}

}

public class Program

{

public static void Main()

{

// Creating a new Person object using the constructor

Person person1 = new Person("Alice", 30);

// Calling the PrintInfo method to display the person's information

person1.PrintInfo();

// Using the SetInfo method to change the person's information

person1.SetInfo("Bob", 25);

person1.PrintInfo();

Console.ReadKey();

}

}

## Example 2 :

using System;

using System.Collections.Generic;

public class Student

{

public int Id { get; }

public string Name { get; }

private List<(string Subject, float TotalMarks, float ObtainedMarks)> marksList = new List<(string, float, float)>();

public Student(int id, string name)

{

Id = id;

Name = name;

}

public void AddMarks(string subject, float totalMarks, float obtainedMarks)

{

marksList.Add((subject, totalMarks, obtainedMarks));

}

public void PrintResult()

{

Console.WriteLine($"Student ID: {Id}, Name: {Name}");

float totalMarksObtained = 0;

float totalMaxMarks = 0;

foreach (var (subject, totalMarks, obtainedMarks) in marksList)

{

totalMarksObtained += obtainedMarks;

totalMaxMarks += totalMarks;

Console.WriteLine($"Subject: {subject}, Total Marks: {totalMarks}, Obtained Marks: {obtainedMarks}");

}

float percentage = (totalMarksObtained / totalMaxMarks) \* 100;

Console.WriteLine($"Total Marks Obtained: {totalMarksObtained}/{totalMaxMarks}");

Console.WriteLine($"Percentage: {percentage}%");

// Calculate grade based on percentage

char grade = 'F'; // Default

if (percentage >= 90)

{

grade = 'A';

}

else if (percentage >= 80)

{

grade = 'B';

}

else if (percentage >= 70)

{

grade = 'C';

}

else if (percentage >= 60)

{

grade = 'D';

}

Console.WriteLine($"Grade: {grade}");

}

}

public class Program

{

public static void Main()

{

Console.Write("Enter student ID: ");

int id = int.Parse(Console.ReadLine());

Console.Write("Enter student name: ");

string name = Console.ReadLine();

Student student = new Student(id, name);

Console.WriteLine("Enter subject name, total marks, and obtained marks (e.g., Math 100 85.5), 'q' to finish:");

string input;

while ((input = Console.ReadLine()) != "q")

{

string[] inputParts = input.Split(' ');

if (inputParts.Length != 3 || !float.TryParse(inputParts[1], out float totalMarks) || !float.TryParse(inputParts[2], out float obtainedMarks))

{

Console.WriteLine("Invalid input format. Please enter subject name, total marks, and obtained marks separated by space.");

continue;

}

string subject = inputParts[0];

student.AddMarks(subject, totalMarks, obtainedMarks);

}

Console.WriteLine();

student.PrintResult();

Console.ReadKey();

}

}