SDLC : System Development Life Cycle

1. **Problem Identification MOST CRITICAL PHASE**

What is the requirement

2. Anaylse the prv or existing system to

understand the problmes

3. Get some alternatives :

We select one solution : Feasibility Study sol1 sol2 sol3 sol4

4. analyse the proposed system

5. Design system > Logical // Physical

6. Test

7. Implement / Give it to the client

8. Post-Implementaion Review

9. Maintainance LONGEST PHASE

Language > Mode of communcation between 2 parties

Every language has

Its own grammar / syntax > Noun pronoun verb

Its own character set > A-Z

COMMON IN ALL LANGUAGES > LOGIC

Programming language > Mode of communication between computer and application

C C++ Java C# PHP

Every language has

Its own grammar / syntax > Every statement has to be terminated by ;

Its own character set > do while for foreach

> Loops Conditions Arrays Functions

COMMON IN ALL LANGUAGES > LOGIC

Program > Statements written to direct / communicate something to system

Which Programming Approach shud be used?

1. Procedural : Follow the steps

C Language

1. Students will come for enqiry

2. Counsellor will handle the enqiury

3. Stydent will take admission

4. Trainer will take session

5.

6.

7.

8.

9.

---

20.

2. Object Oriented

C++ , Java , C# , Python

Counsellor > Handle Enquiry , Give Admission , Give Certificate ,

Student > Come for Enquiry, Take Admission,

Trainer > Take Classes Give Test,

Trainer > Take sesions , Give Test, Check test

Concepts of OOPs

Class > Blueprint or user defined type based on which multiple objects of same type could be created

Object >Instance of a class

If Fruit is a class , apple mango are the objects

Encapsulation > Binding of data members & methods within a class . Hiding the details of a class which is not needed

Abstraction > Showing the details that the users needs

How do we achieve them > By using access specifiers , private public internal protected protected internal

Polymorphism > One name different forms .

It is of 2 types

Static / Compile Time poly > By method overloading

Run time / Dynamic Poly > By method overriding & Virtual functions

Inheritance > Deriving features of one class to other. Reusability

Constraints > Restrictions that what values are allowed in the columns of table

1. Primary Key, it has 2 features , unique , not null
2. Unique > How many nulls are allowed > on1
3. Foreign key
4. Not null
5. Check
6. Default

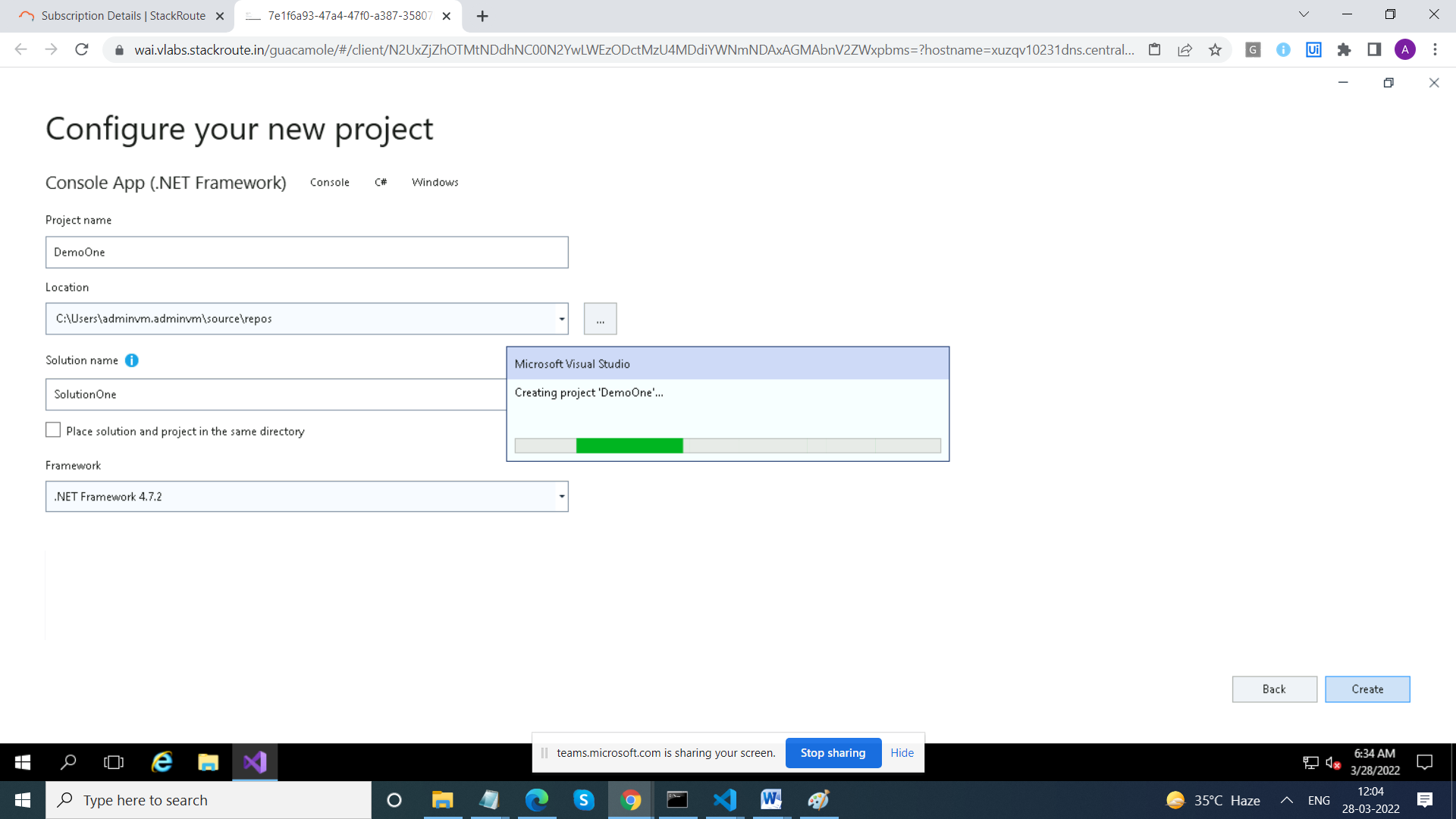
How many P Key we can have in a table > 1

How many F Key we can have in a table > More than 1

To write your programs in C# , we need some editor

Visual Studio : IDE , Integrated Development Environment

C:\Users\Anamika\source\repos



Solution : Container for Multiple Projects

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FirstProject

{

class Program

{

static void Main(string[] args)

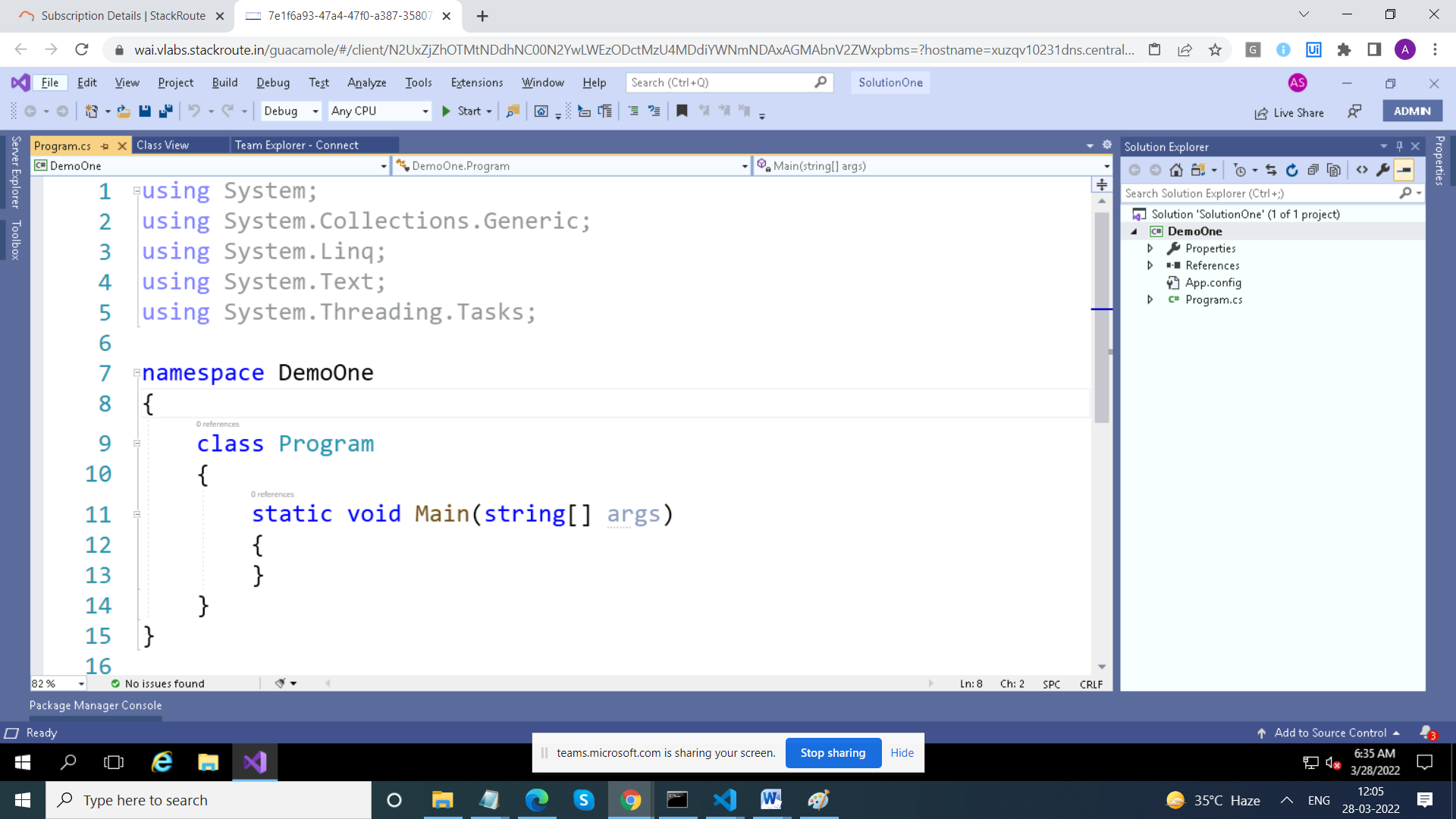
{

}

}

}

// System, System.Collections : These are namespaces



using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

Console.Write("Hello"); // Console is a class

// Console class is in System namespace

// Write is Method of Console Class

// Namespace : Its a collectionn of logically related

// classes or namespaces

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DemoOne

{

class Program

{

static void Main(string[] args)

{

// Console.Write("hello1");

// Console.Write("hello2");

// //Console.Write("hello3");

Console.WriteLine("hello1");

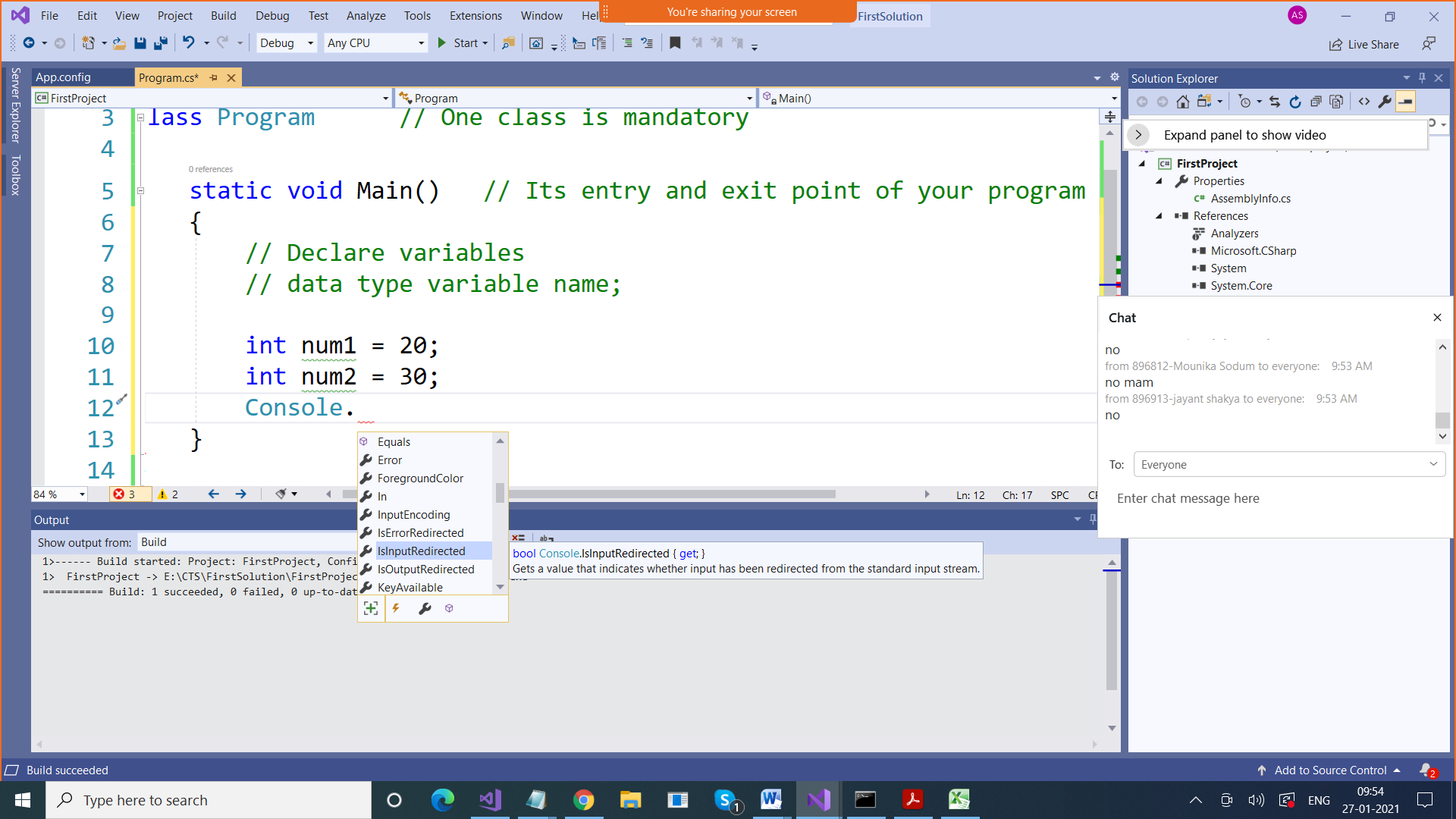
Console.WriteLine("hello2");

Console.WriteLine("hello3");

}

}

}



Intellisense : We get Suggestions

int num = 20;

Console.WriteLine("Value of num1 is" + num);

Console.WriteLine("Value of num1 is {0}" , num);

Console.WriteLine($"Value of num1 is {num}");

// Get sum of 2 numbers

int num1 = 10;

int num2 = 20;

Console.WriteLine("Sum of num1 & num2 is " + (num1+num2));

Console.WriteLine("Sum of {0} & {1} is {2}", num1, num2, num1 + num2);

Console.WriteLine($"Sum of {num1} & {num2} is {num1+ num2}");

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

// Declare variables

// data type variable name;

int num1 = 20;

int num2 = 30;

Console.Write(num1 + num2);

}

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

// Declare variables

// data type variable name;

int num1 = 10;

int num2 = 40;

Console.WriteLine(num1 + num2);

Console.WriteLine("Sum of 20 amd 30 is " + ( num1 + num2));

Console.WriteLine("Sum of "+ num1 + " and " + num2 + " is " + (num1 + num2));

}

}

// Using Positional parameters

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

// Declare variables

// data type variable name;

int num1 = 10;

int num2 = 40;

Console.WriteLine(num1 + num2);

Console.WriteLine("Sum of 20 amd 30 is " + ( num1 + num2));

Console.WriteLine("Sum of "+ num1 + " and " + num2 + " is " + (num1 + num2));

// printf("%d + %d = %d", num1 , num2 , num1+num2);

Console.WriteLine("Sum of {0} and {1} is {2}" ,

num1 , num2, (num1 + num2));

// Here {0} , {1} and {2} are known as positional parameters

}

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

// Declare variables

// data type variable name;

int num1 = 10;

int num2 = 40;

Console.WriteLine("Sum of {0} and {1} is {2}" ,

num1 , num2, (num1 + num2));

Console.WriteLine("Difference of {0} and {1} is {2}",

num1, num2, (num1 - num2));

Console.WriteLine("Product of {0} and {1} is {2}",

num1, num2, (num1 \* num2));

Console.WriteLine("Remainder of {0} and {1} is {2}",

num1, num2, (num1 % num2));

Console.WriteLine("Quotient of {0} and {1} is {2}",

num1, num2, (num1 / num2));

}

}

Console.ReadLine(); > To Read values at run time

It takes input in string form, we need to convert it to type that we want

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

// Declare variables

// data type variable name;

int num1,num2;

Console.WriteLine("Enter Value of Num1");

num1 = Convert.ToInt16( Console.ReadLine());

Console.WriteLine("Enter Value of Num2");

num2 = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Sum of {0} and {1} is {2}" ,

num1 , num2, (num1 + num2));

Console.WriteLine("Difference of {0} and {1} is {2}",

num1, num2, (num1 - num2));

Console.WriteLine("Product of {0} and {1} is {2}",

num1, num2, (num1 \* num2));

Console.WriteLine("Remainder of {0} and {1} is {2}",

num1, num2, (num1 % num2));

Console.WriteLine("Quotient of {0} and {1} is {2}",

num1, num2, (num1 / num2));

}

}

Statements is of 3 types in any programming language

1. Sequential :

int num1,num2;

Console.WriteLine("Enter Value of Num1");

num1 = Convert.ToInt16( Console.ReadLine());

Console.WriteLine("Enter Value of Num2");

num2 = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Sum of {0} and {1} is {2}" ,

num1 , num2, (num1 + num2));

Console.WriteLine("Difference of {0} and {1} is {2}",

num1, num2, (num1 - num2));

Console.WriteLine("Product of {0} and {1} is {2}",

num1, num2, (num1 \* num2));

Console.WriteLine("Remainder of {0} and {1} is {2}",

num1, num2, (num1 % num2));

Console.WriteLine("Quotient of {0} and {1} is {2}",

num1, num2, (num1 / num2));

2. Conditional Statements

We use

* If
* If else
* If else if ,else
* switch

**if Syntax**

if(condition)

{

Statements;

}

**If else Syntax**

if(condition)

{

Statements;

}

Else

{ statements;}

**If else if Syntax**

if(condition1)

{

Statements;

}

Else if(condition2)

{ statements;}

Else

{

Statements;

}

**swicth Syntax**

switch(expression)

{

case value: {Statements; break;}

case value: {Statements; break;}

case value: {Statements; break;}

default: {Statements; break;}

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

// Declare variables

// data type variable name;

int num1,num2;

int ch;

Console.WriteLine("Enter Value of Num1");

num1 = Convert.ToInt16( Console.ReadLine());

Console.WriteLine("Enter Value of Num2");

num2 = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Enter your choice");

ch = Convert.ToInt16( Console.ReadLine());

if(ch==1)

Console.WriteLine("Sum of {0} and {1} is {2}" ,

num1 , num2, (num1 + num2));

else if(ch==2)

Console.WriteLine("Difference of {0} and {1} is {2}",

num1, num2, (num1 - num2));

else if(ch==3)

Console.WriteLine("Product of {0} and {1} is {2}",

num1, num2, (num1 \* num2));

else if(ch==4)

Console.WriteLine("Remainder of {0} and {1} is {2}",

num1, num2, (num1 % num2));

else if(ch==5)

Console.WriteLine("Quotient of {0} and {1} is {2}",

num1, num2, (num1 / num2));

else

Console.WriteLine("Invalid choice");

}

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

// Declare variables

// data type variable name;

int num1, num2;

int ch;

Console.WriteLine("Enter Value of Num1");

num1 = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Enter Value of Num2");

num2 = Convert.ToInt16(Console.ReadLine());

Console.WriteLine("Enter your choice");

ch = Convert.ToInt16(Console.ReadLine());

switch (ch)

{

case 1:

Console.WriteLine("Sum of {0} and {1} is {2}",

num1, num2, (num1 + num2));

break;

case 2:

Console.WriteLine("Difference of {0} and {1} is {2}",

num1, num2, (num1 - num2));

break;

case 3:

Console.WriteLine("Product of {0} and {1} is {2}",

num1, num2, (num1 \* num2));

break;

case 4:

Console.WriteLine("Remainder of {0} and {1} is {2}",

num1, num2, (num1 % num2));

break;

case 5:

Console.WriteLine("Quotient of {0} and {1} is {2}",

num1, num2, (num1 / num2));

break;

default:

Console.WriteLine("Invalid choice");

break;

}

}

}

3.Repetitive Statements (Loops)

* While
* Do while
* For
* foreach

In a loop , we have 3 statements

1. Initialization part (Starting point)
2. Conditional /End point
3. Increment / Decrement State

Do – while loop

Initialization part;

Do

{

Statements;

Ince / dece statement;

} while(condition)

while loop

Initialization part;

While(condition)

{

Statements;

Ince / dece statement;

}

for loop

for(Initialization part;condition; Ince / dece statement;)

{

Statements;

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

int num = 1;

do

{

Console.Write(num + " " );

num++;

}

while (num <= 10);

}

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

int num = 1;

while(num<=10)

{

Console.Write(num + " " );

num++;

}

}

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

for(int num=1; num<=10; num++)

{

Console.Write(num + " " );

}

}

}

using System; // namespace

class Program // One class is mandatory

{

static void Main() // Its entry and exit point of your program

{

for(int num=2; num<=10; num+=2)

{

Console.Write(num + " " );

}

}

}