Demo1:

namespace ConsoleApp1\_pre

{

internal class Program

{

static void Main(string[] args)

{

int a, b, c;

Console.WriteLine("Enter first number");

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

Console.WriteLine("Enter second number");

b = Convert.ToInt32(Console.ReadLine());

c = a + b;

// Console.WriteLine("Addition:" + c);

// Console.WriteLine("Addition:{0}", c);

// Console.WriteLine($"Addition:{c}" );

//Console.WriteLine("Addition of {0} and {1}={2}", a,b,c);

Console.WriteLine($"Addition of {a} and {b}={c}");

Console.ReadKey();

}

}

}

Demo 2:

namespace ConsoleApp2\_pre

{

internal class Program

{

static void Main(string[] args)

{

int rno, m1, m2, m3, tot;

string sname;

double per;

Console.WriteLine("Enter Rollno:");

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

Console.WriteLine("Enter Name");

sname=Console.ReadLine();

Console.WriteLine("Enter marks of 3 subjects");

m1 = int.Parse(Console.ReadLine());

m2 = int.Parse(Console.ReadLine());

m3 = int.Parse(Console.ReadLine());

tot = m1 + m2 + m3;

per = (double)tot / 3;

Console.WriteLine("Percentage:" + per);

if (per >= 60)

Console.WriteLine("First Class");

else

if (per >= 50)

Console.WriteLine("Second Class");

else

if (per >= 35)

Console.WriteLine("Pass");

else

Console.WriteLine("Fail");

Console.ReadKey();

}

}

}

**Namespace demo**

namespace first

{

public class Student

{

public void show()

{

Console.WriteLine("first student show");

}

}

}

namespace second

{

public class Student

{

public void display()

{

Console.WriteLine("second Student display");

}

}

}

Program.cs

using first;

using second;

namespace namespacedemo\_pre

{

internal class Program

{

static void Main(string[] args)

{

first.Student s= new first.Student();

s.show();

second.Student s2= new second.Student();

s2.display();

Console.ReadKey();

}

}

}

**Constantdemo:**

namespace constantdemo\_pre

{

internal class Program

{

static void Main(string[] args)

{

const int x = 5;

const int num = x \* 3;

Console.WriteLine(num);

Console.ReadKey();

}

}

}

//constant variable value cannot be change once it is assigned

**Enum:**group of integer constant

Enumdemo:

namespace enumdemo\_pre

{

enum months { jan=1,feb,mar,april,may,june};

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine(months.mar);

Console.WriteLine((int)months.mar);

Console.ReadKey();

}

}

}

**Boxing and Unboxing**

//boxing:value type is stored in reference type,implicit conversion

//unboxing:reference type is stored in value type,need explicit conversion

namespace boxingdemo\_pre

{

internal class Program

{

static void Main(string[] args)

{

int a = 10;

object o = a; //boxing

Console.WriteLine(o);

int b = (int)o; //unboxing

Console.WriteLine(b);

Console.ReadKey();

}

}

}

Arraydemo1:

namespace arraydemo\_pre

{

internal class Program

{

static void Main(string[] args)

{

int[] a = new int[5];

Console.WriteLine("Enter 5 numbers");

for (int i = 0; i < a.Length; i++)

{

a[i]=int.Parse(Console.ReadLine());

}

Console.WriteLine("\nDisplay Array");

for (int i = 0; i < a.Length; i++)

{

Console.WriteLine(a[i]);

}

Console.ReadKey();

}

}

}

**Array demo2:**

namespace arraydemo2\_pre

{

internal class Program

{

static void Main(string[] args)

{

int[] a = { 10, 20, 30, 40, 50 };

Console.WriteLine("\nDisplay Array");

for (int i = 0; i < a.Length; i++)

{

Console.WriteLine(a[i]);

}

Console.ReadKey();

}

}

}

**Array demo 3:**

//for loop-updatable(insert/delete)

//for each:read-only

namespace arraydemo2\_pre

{

internal class Program

{

static void Main(string[] args)

{

int[] a = { 10, 20, 30, 40, 50 };

Console.WriteLine("\n----------------Using for loop------------------");

for (int i = 0; i < a.Length; i++)

{

Console.WriteLine(a[i]);

}

Console.WriteLine("\n-----------------using foreach loop---------------");

foreach (int n in a)

{

Console.WriteLine(n);

}

Console.ReadKey();

}

}

}

**Method demo 1:**

//no return type,no parameters

namespace methoddemo\_pre

{

class Accenture

{

public void addition()

{

int a, b, c;

Console.WriteLine("Enter 2 numbers:");

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

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

c = a + b;

Console.WriteLine("Addition:" + c);

}

}

internal class Program

{

static void Main(string[] args)

{

Accenture obj=new Accenture();

obj.addition();

Console.ReadKey();

}

}

}

**Method demo2:**

//no return type,with parameters

namespace methoddemo2\_pre

{

class Accenture

{

public void addition(int a,int b)

{

int c = a + b;

Console.WriteLine("Addition:" + c);

}

}

internal class Program

{

static void Main(string[] args)

{

int x,y;

Console.WriteLine("Enter 2 numbers:");

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

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

Accenture obj = new Accenture();

obj.addition(x,y);

Console.ReadKey();

}

}

}

**Method demo 3**

//with return type,with parameters

namespace methoddemo3\_pre

{

class Accenture

{

public int addition(int a, int b)

{

return a + b;

}

}

internal class Program

{

static void Main(string[] args)

{

int x, y,c=0;

Console.WriteLine("Enter 2 numbers:");

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

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

Accenture obj = new Accenture();

c=obj.addition(x, y);

Console.WriteLine("Addition:" + c);

Console.ReadKey();

}

}

}

**Call by value**

//call by value

namespace callbyval\_pre

{

class Accenture

{

public void swap(int x,int y)

{

int temp;

temp = x;

x = y;

y = temp;

}

}

internal class Program

{

static void Main(string[] args)

{

Accenture obj= new Accenture();

int a, b;

a = 10;

b = 20;

obj.swap(a,b);

Console.WriteLine("After swap:a:{0} and b={1}",a,b);

Console.ReadKey();

}

}

}

**Call by reference**

//call by reference

namespace call\_by\_ref\_pre

{

class Accenture

{

public void swap(ref int x,ref int y)

{

int temp;

temp = x;

x = y;

y = temp;

}

}

internal class Program

{

static void Main(string[] args)

{

Accenture obj = new Accenture();

int a, b;

a = 10;

b = 20;

obj.swap(ref a,ref b);

Console.WriteLine("After swap:a:{0} and b={1}", a, b);

Console.ReadKey();

}

}

}

Out parameter demo

namespace outdemo\_pre

{

class Accenture

{

public void show(int a, int b,out int c,out int d)

{

c=a+b;

d = a - b;

}

}

internal class Program

{

static void Main(string[] args)

{

Accenture obj = new Accenture();

int x, y,z,p;

x = 80;

y = 30;

obj.show(x, y,out z,out p);

Console.WriteLine("Addition:" + z);

Console.WriteLine("Subtraction:" + p);

Console.ReadKey();

}

}

}

Optional demo

namespace optionaldemo\_pre

{

class Accenture

{

public void show(int a,int b=100)

{

Console.WriteLine("a:" + a);

Console.WriteLine("b:" + b);

}

}

internal class Program

{

static void Main(string[] args)

{

Accenture obj=new Accenture();

obj.show(50);

Console.ReadKey();

}

}

}

**Name parameter demo**

namespace namedemo\_pre

{

class Accenture

{

public void show(int a, int b)

{

Console.WriteLine("a:" + a);

Console.WriteLine("b:" + b);

}

}

internal class Program

{

static void Main(string[] args)

{

Accenture obj = new Accenture();

obj.show(b:50,a:100);

Console.ReadKey();

}

}

}

**Params parameter demo**

namespace paramdemodemo\_pre

{

class Accenture

{

public void show(int n,params string []s)

{

foreach (string o in s)

{

Console.WriteLine(o);

}

}

}

internal class Program

{

static void Main(string[] args)

{

Accenture obj = new Accenture();

obj.show(100,"sahil","ram","pooja","soham","rohan");

Console.ReadKey();

}

}

}