**using** System;

**class** Program

{

**static** **void** Main()

{

**string** s1 = "Good";

**string** s2 ="Morning";

**string** s3=**string**.Concat(s1, s2);

Console.WriteLine(s3);

Console.ReadLine();

}

}

**using** System;

**public** **class** Program

{

**public** **static** **void** Main()

{

**string** str;

Console.WriteLine("Enter the String in Uppercase :");

str = Console.ReadLine();

Console.WriteLine("String in LowerCase : {0}", str.ToLower());

Console.ReadLine();

}

}

using System;

public class ArrayExample

{

public static void Main(string[] args)

{

int[] arr = new int[5];//creating array

arr[0] = 10;//initializing array

arr[2] = 20;

arr[4] = 30;

//traversing array

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

{

Console.WriteLine(arr[i]);

}

}

}

using System;

public class ArrayExample

{

static void printArray(int[] arr)

{

Console.WriteLine("Printing array elements:");

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

{

Console.WriteLine(arr[i]);

}

}

public static void Main(string[] args)

{

int[] arr1 = { 25, 10, 20, 15, 40, 50 };

int[] arr2 = { 12, 23, 44, 11, 54 };

printArray(arr1);//passing array to function

printArray(arr2);

}

}

using System;

public class ArrayExample

{

static void printMin(int[] arr)

{

int min = arr[0];

for (int i = 1; i < arr.Length; i++)

{

if (min > arr[i])

{

min = arr[i];

}

}

Console.WriteLine("Minimum element is: " + min);

}

public static void Main(string[] args)

{

int[] arr1 = { 25, 10, 20, 15, 40, 50 };

int[] arr2 = { 12, 23, 44, 11, 54 };

printMin(arr1);//passing array to function

printMin(arr2);

}

}

using System;

public class MultiArrayExample

{

public static void Main(string[] args)

{

int[,] arr=new int[3,3];//declaration of 2D array

arr[0,1]=10;//initialization

arr[1,2]=20;

arr[2,0]=30;

//traversal

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

Console.Write(arr[i,j]+" ");

}

Console.WriteLine();//new line at each row

}

}

}

using System;

namespace CallByValue

{

class Program

{

// User defined function

public void Show(int val)

{

val \*= val; // Manipulating value

Console.WriteLine("Value inside the show function "+val);

// No return statement

}

// Main function, execution entry point of the program

static void Main(string[] args)

{

int val = 50;

Program program = new Program(); // Creating Object

Console.WriteLine("Value before calling the function "+val);

program.Show(val); // Calling Function by passing value

Console.WriteLine("Value after calling the function " + val);

}

}

}

using System;

public class Student

{

int id;//data member (also instance variable)

String name;//data member(also instance variable)

public static void Main(string[] args)

{

Student s1 = new Student();//creating an object of Student

s1.id = 101;

s1.name = "Sonoo Jaiswal";

Console.WriteLine(s1.id);

Console.WriteLine(s1.name);

}

}

using System;

public class Student

{

public int id;

public String name;

public void insert(int i, String n)

{

id = i;

name = n;

}

public void display()

{

Console.WriteLine(id + " " + name);

}

}

class TestStudent{

public static void Main(string[] args)

{

Student s1 = new Student();

Student s2 = new Student();

s1.insert(101, "Ajeet");

s2.insert(102, "Tom");

s1.display();

s2.display();

}

}

using System;

public class Employee

{

public int id;

public String name;

public float salary;

public void insert(int i, String n,float s)

{

id = i;

name = n;

salary = s;

}

public void display()

{

Console.WriteLine(id + " " + name+" "+salary);

}

}

class TestEmployee{

public static void Main(string[] args)

{

Employee e1 = new Employee();

Employee e2 = new Employee();

e1.insert(101, "Sonoo",890000f);

e2.insert(102, "Mahesh", 490000f);

e1.display();

e2.display();

}

}

using System;

public class Employee

{

public int id;

public String name;

public float salary;

public Employee(int i, String n,float s)

{

id = i;

name = n;

salary = s;

}

public void display()

{

Console.WriteLine(id + " " + name+" "+salary);

}

}

class TestEmployee{

public static void Main(string[] args)

{

Employee e1 = new Employee(101, "Sonoo", 890000f);

Employee e2 = new Employee(102, "Mahesh", 490000f);

e1.display();

e2.display();

}

}

# C# Destructor

A destructor works opposite to constructor, It destructs the objects of classes. It can be defined only once in a class. Like constructors, it is invoked automatically.

using System;

public class Employee

{

public Employee()

{

Console.WriteLine("Constructor Invoked");

}

~Employee()

{

Console.WriteLine("Destructor Invoked");

}

}

class TestEmployee{

public static void Main(string[] args)

{

Employee e1 = new Employee();

Employee e2 = new Employee();

}

}