

Smit Vora

Demo Practicals

Module-2

Topics

Prerequisite to run code

```
namespace Application
{
    class Program
    {
        static void Main(string[] args)
        {
            //demo code
        }
    }
}
```

9. Loop Iteration

9.1. while loop example

- `int i=0;`
`while(i<5)`
`{`
`Console.WriteLine(i);`
`i++;`
`}`

`//result – 0,1,2,3,4`

9.2. do...while loop example

- `int i = -4;`

```
do
{
    Console.WriteLine(i);
    i++;
}
while(i > 0);

//result – -4 as even I doesn't meet condition loop is excuted atleast one
time.
```

9.3. for loop example

- ```
for(int i = 0; i <= 10; i = i + 2)
{
 Console.WriteLine(i);
}

//result – 0,2,4,6,8,10
```

### 9.4. foreach loop example

- ```
string [ ] fruits = {"mango","apple","guava","banana"}
foreach(string fruit in fruits)
{
    Console.WriteLine(fruit);
}

//result – mango, apple, guava, banana
```

10. Understanding Arrays

10.1. Array example

```
int[] arr = { 10, 20, 30, 40, 50 };//creating and initializing array

int sum=0;

//traversing array
foreach (int i in arr)
{
    sum+=i;
}

Console.WriteLine("Total sum = {0}",sum);

//result = Total sum = 150
```

11. Defining and Calling Methods

11.1. different type of parameters in method (Value type, Ref. type, optional)

- Value type parameter example

```
using System;

public class Mul {

    static public void Main()

    {    // The value of the parameter

        // is already assigned

        int a = 4

        int b = 5

        int res = multiply(a, b);

        Console.WriteLine(res);

    }
```

```

        public static int multiply(int n1, int n2)
        {
            return n1 * n2;
        }
    }

```

//result – 20

- **Reference type example**

```
using System;
```

```
class Matching {
```

```
    public static void Main()
```

```
    {        // Assigning value
```

```
        string val = "Smit Vora";
```

```
        // Pass as a reference parameter
```

```
        CompareValue(ref val);
```

```
        Console.WriteLine(val);
```

```
    }
```

```
static void CompareValue(ref string val1)
```

```
{
```

```
    // Compare the value
```

```
    if (val1 == " Smit Vora ")
```

```
    {
```

```

        Console.WriteLine("Matched!");
    }

    // Assigning new value
    val1 = "Not Me";
}

}

// result – Matched!

    Not Me

// as change made in method reflected on the variable in the main method
through reference.

```

- **Optional or Default type**

using System;

```

public class Operation {

    // operating contain three parameters
    public static void operating (int a, int b, int c = 2)
    {

        int result = a+b*c;

        Console.WriteLine("Final result is: " + result);

    }

    // Main Method
    static public void Main()
    {

```

```

        // calling the static method with named
        // parameters without any order
        operating (6,2);

    }
}

```

//result - Final result is: 10

As default value of c is taken as 2

- **Named parameters example**

using System;

```

public class Operation {

    // operating contain three parameters
    public static void operating (int a, int b, int c)
    {

        int result = a+b*c;

        Console.WriteLine("Final result is: " + result);

    }

    // Main Method
    static public void Main()
    {

        // calling the static method with named
        // parameters without any order
        operating (b: 6, a: 2, c: -2);
    }
}

```

```

    }

}

//result - Final result is: -10

```

- **Out parameter example**

```
using System;
```

```
class Matching {
```

```
    public static void Main()
```

```
    {        // Assigning value
```

```
        string val ;
```

```
        // Pass as a reference parameter
```

```
        ChangeValue (out val);
```

```
        Console.WriteLine(val);
```

```
    }
```

```
static void ChangeValue(out string val1)
```

```
{
```

```
    // Compare the value
```

```
    if (val1 == " Smit Vora ")
```

```
    {
```

```
        Console.WriteLine("My Name!");
```

```
    }
```

```
// Assigning new value
val1 = "Not Me";
}
}

// result – My Name!

Not Me

// as change made in method reflected on the variable in the main method
through reference and variable just need to be declared not assigned so
different from reference.
```

12. Working with strings

12.1. Use of Various methods in String class

```
string a = "hello";
string b = "world";
string c = (String)a.Clone();
string d = string.Concat(a,b);
string e = string.Copy(d);
string f = a.Remove(2);
string g = b.Replace("wor","smit");
Console.WriteLine(c);
Console.WriteLine(d);
Console.WriteLine(e);
Console.WriteLine(f);
Console.WriteLine(g);
```



```
//result – hello
    helloworld
    helloworld
    he
    smitld
```

13. Working with Datetime

13.1. Properties of Datetime

```
DateTime DateTimeProperty = new DateTime(2021, 9, 17, 7, 10, 24);
Console.WriteLine("Day:{0}", DateTimeProperty.Day);
Console.WriteLine("Month:{0}", DateTimeProperty.Month);
Console.WriteLine("Year:{0}", DateTimeProperty.Year);
Console.WriteLine("Hour:{0}", DateTimeProperty.Hour);
Console.WriteLine("Minute:{0}", DateTimeProperty.Minute);
Console.WriteLine("Second:{0}", DateTimeProperty.Second);
Console.WriteLine("Millisecond:{0}", DateTimeProperty.Millisecond);
Console.WriteLine("Day of Week:{0}", DateTimeProperty.DayOfWeek);
Console.WriteLine("Day of Year: {0}", DateTimeProperty.DayOfYear);
Console.WriteLine("Time of Day:{0}", DateTimeProperty.TimeOfDay);
```

//result - Day:17
 Month:9
 Year:2021
 Hour:7
 Minute:10
 Second:24
 Millisecond:0
 Day of Week:Friday
 Day of Year: 260
 Time of Day:07:10:24