String

- A string is basically a **sequence of characters**
- A **string** in **C#** is an object of type String The string type represents a string of **Unicode Characters**. String objects are **immutable** that is **they cannot be changed after they have been created.**

Example

Enter your optional string in c# and print it ? Try enter no.

```
string ST;
Console.WriteLine("Enter your Message :");
ST = Console.ReadLine();
Console.WriteLine("The String is {0}.", ST);
Console.ReadKey();
```

Example

Enter your optional string in c# and print each char in it ?

```
string S;
 Console.WriteLine("Enter your Message :");
 S = Console.ReadLine();
 Console.WriteLine(S.Length );
 for (int i = 0; i < S.Length; i++)
  Console.WriteLine(S[i]);
  Console.ReadKey();
```

Example

Enter your Message in c# using for loop and special string for stop loop?

```
for (int i = 0; i < 5; i++)
{
Console.WriteLine("Enter your Message and when you need to stop enter  exit");
String line = Console.ReadLine();
  if (line == "exit")
        break;
   else Console.WriteLine(line.Length);
            }      Console.ReadKey();
```

Example

Print the inverse of string?

```
string Message = "Welcome Dear ";
for (int i = 0; i < Message.Length; i++)
Console.Write(Message[Message.Length - i - 1]);     //try WriteLine
Console.WriteLine();
Console.WriteLine (Message);
Console.ReadKey();
```

Example

Print the number of words in the string?

```
String s = "hello dear how are you what did do yesterday";
int m = 1;
for (int i = 1; i < s.Length; i++)
    if (s[i] == ' ')
        m = m + 1;
Console.WriteLine(m);
Console.ReadKey();
```

Example

Print the char after char ‘d’ in the string?

```
String s = "hello dear how are you what did do yesterday";
 for (int i = 1; i < s.Length; i++)
    if (s[i] == 'd')
        Console.WriteLine(s[i+1]);
Console.ReadKey();
```

Example

Using Foreach

- This is the easiest, loop.
- Foreach uses no integer index. it returns each element in order.

```
string s = "Computer Science";
 foreach (char c in s)
     Console.WriteLine(c);
      Console.ReadKey();
```

String methods - String.Concat

With **concat** two or more strings become one. It is possible to concatenate two or more strings with several syntax forms. We use the plus operator and the string.Concat method. The plus compiles into string.Concat.

String methods - String.Concat

```
string s1 = "Hello ";
 string s2 = s1;
 s1 += "World";
 System.Console.WriteLine(s2);
 //Output: Hello
 Console.ReadKey();
```

String methods - String.Concat

write **C# program that concatenates two strings**

```
string s1 = "string2";
string s2 = string.Concat("string1", s1);
Console.WriteLine(s2);
Console.ReadKey();
```

String methods - String.Concat

write **C# program that concatenates three strings**

```
string s1 = "string2";
string s2 = "string2";
string s3 = s1 + s2 + "string3";
Console.WriteLine(s3);
Console.ReadKey();
```

String methods - String.Concat

C# program that concats string list

```
var list = new List<string>();
list.Add("cat");
list.Add("dog");
list.Add("perls");
string M = string.Concat(list);
Console.WriteLine(M);
Console.ReadKey();
```

String methods - String.Replace

- **Replace.** A string contains incorrect chars. It has XYZ but we want ABC. Replace helps with this puzzle. It swaps those substrings.
- every replacement made results in a new string.
- **To begin,** we invoke the Replace method. Please note that we must assign Replace's result to a variable. It does not modify the string in-place.

String methods - String.Replace

C# program that uses Replace

```
string input = "_::_pagitica";
Console.WriteLine(input);
string output = input.Replace("_::_", "Areo");
Console.WriteLine(output);
Console.ReadKey();
```

String methods - String.Replace

C# program that causes multiple replacements

```
const string s = "Dot Net Perls is about Dot Net.";
        Console.WriteLine(s);
        string v = s.Replace("Net", "Basket");
        Console.WriteLine(v);
        Console.ReadKey();
```

String methods - String.IndexOf

- **IndexOf.** A string contains many characters. These characters may be searched and tested. We simplify these operations with IndexOf.
- **This method,** a string method, returns the first index of a letter in a string. It can also find a substring. It is often used in looping constructs.
- It returns negative one when nothing is found. We must test for -1 if no value may exist.

String methods - String.IndexOf

C# program that uses IndexOf

```
const string value = "Your dog is cute.";
if (value.IndexOf("dog") != -1)
    Console.WriteLine("string contains dog!");
        Console.ReadKey();
```

String methods - String.Compare

Compare. This method determines the sort order of strings. It checks if one string is ordered before another when in alphabetical order, whether it is ordered after, or is equivalent.

String methods - String.Compare

String Compare method results:

String A: First alphabetically

String B: Second alphabetically

Compare(A, B): -1

Compare(B, A): 1

Compare(A, A): 0

Compare(B, B): 0

String methods - String.Compare

C# program that uses Compare

```
string a = "a";
string b = "b";
int c = string.Compare(a, b);
Console.WriteLine(c);
c = string.Compare(b, a);
Console.WriteLine(c);
c = string.Compare(a, a);
Console.WriteLine(c);
c = string.Compare(b, b);
Console.ReadKey();
```

String methods - String.Contains

Contains. This method searches strings. It checks if one substring is contained in another. Contains returns true or false, not an index.

String methods - String.Contains

C# program that uses Contains

```
bool y;
string x = "Hello dear How are you now?";
y = x.Contains("dear");
 Console.WriteLine(y);
  Console.ReadKey();
```

String methods - String.Contains

C# program that uses Contains

```
string x = "Hello dear How are you now?";
if (x.Contains("dear"))
Console.WriteLine("Good news" );
Console.ReadKey();
```

Name Example

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    String FirstName= "Ali" ;
    string MiddleName = "Hayder";
    string LastName = "Hassan";
    String FullName = FirstName + MiddleName + LastName;
    string txt = FullName.Remove(3, 6 );
    textBox1.Text = txt;
    textBox2.Text = txt.Insert(3, MiddleName);
```

String.ToUpper/ ToLower

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    String FirstName= "Ali" ;
    string MiddleName = "Hayder";
    string LastName = "Hassan";
    String FullName = FirstName + MiddleName + LastName;
    string txt = FullName.Remove(3, 6 );
    textBox1.Text = txt;
    textBox2.Text = txt.Insert(3, MiddleName);
    textBox3.Text = txt.ToUpper();
    textBox4.Text = txt.ToLower ();


}
```

C# String Substring

How to use C# string Substring

Substring in C# string Class returns a new string that is a substring of this string. The substring begins at the specified given index and extended up to the given length.

string string.substring(int startIndex,int length)

Parameters:

startIndex: The index of the start of the substring.

length: The number of characters in the substring.

Returns:

The specified substring.

C# String Substring

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string s = null;
    string ss = null;
    s= "This is substring test";
   ss= s.Substring(8, 9);
    MessageBox.Show(ss);
}
```

C# String Index of

How to use C# string IndexOf

The **IndexOf** method in string Class in C# returns the index of the first occurrence of the specified substring.

int string.IndexOf(string str)

Parameters:

str - The parameter string to check its occurrences

Returns:

Integer - If the parameter String occurred as a substring in the specified String

it returns position of the first character of the substring .

If it does not occur as a substring, -1 is returned.

C# String Index of

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
     String str = "CSharp TOP 10 BOOKS";
    textBox1 .Text = (str.IndexOf("TOP").ToString());


     }
```

C# String Contains

How to use C# string Contains

The CSharp **Contains** method returns true if and only if this string contains the specified sequence of char values.

bool string.containe(string str)

Parameters:

String str - input String for search

Returns:

Boolean - Yes/No

If the str Contains in the String then it returns true

If the str does not Contains in the String it returns False

C# String Contains

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string str = null;
    str = "CSharp TOP 10 BOOKS";
    if (str.Contains("TOP") == true)
        MessageBox.Show("The string Contains() 'TOP' ");
    else
        MessageBox.Show("The String does not Contains() 'TOP'");
}
```

C# String Spilt

C# Split() handles splitting upon given string and character delimiters. It returns an array of String containing the substrings delimited by the given System.Char array.

string = "dd-mm-yy"

If your String contains "dd-mm-yy", split on the "-" character to get an array of: "dd" "mm" "yy".

C# String Spilt

Syntax :

string[] string.split(string[] separator)

Parameters:

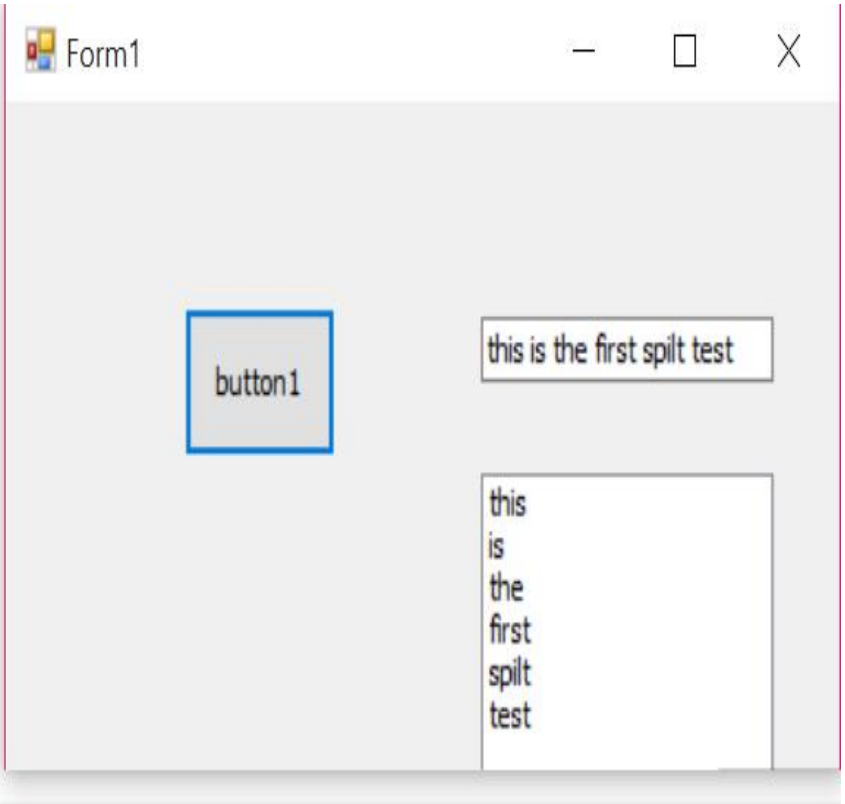
separator - the given delimiter

Returns:

An array of Strings delimited by one or more characters in separator

C# String Spilt

```
    InitializeComponent();
}

1 reference
private void button1_Click(object sender, EventArgs e)
{
    string st1 = textBox1 .Text ;

    char[] splitchar = { ' ' };
    string [] st2 = st1.Split(splitchar);
    for (int count = 0; count < st2.Length - 1; count++)
     listBox1 .Items .Add (st2[count]);
    }
    }
```



C# String Array

- Arrays is Data Structure, are using for store similar data types grouping as a single unit.
- We can access Array elements by its numeric index.
- The array indexes start at zero.
- Array Declaration and initialization

```
string[] week = new string[] {"Sunday","Monday","Tuesday"};
```

- Retrieve a single item from Array.

```
string str = week[1];
```

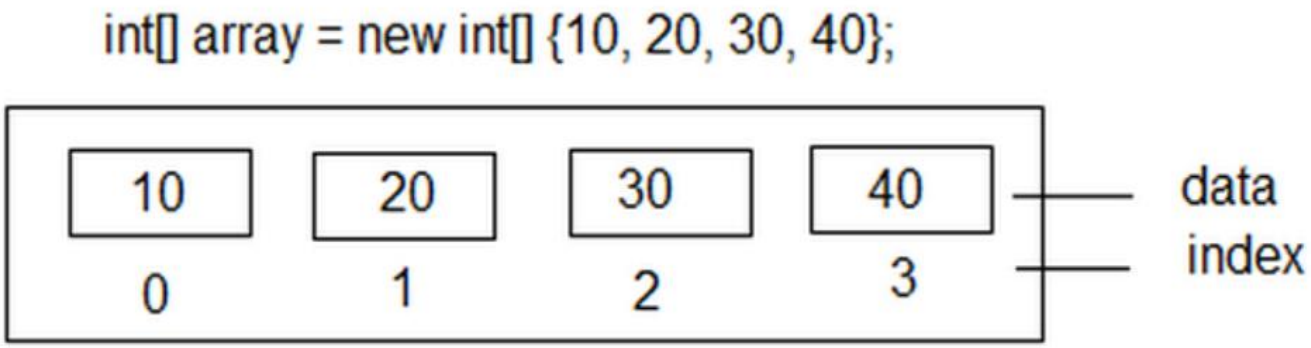
C# Array Examples

- Array Declaration and initialization

string[] week = new string[] {"Sunday","Monday","Tuesday"};

- Retrieve a single item from Array.

string s = week[1];



C# String Array

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string[] week = new string[7];
    week[0] = "Sunday";
    week[1] = "Monday";
    week[2] = "Tuesday";
    week[3] = "Wednsday";
    week[4] = "Thursday";
    week[5] = "friday";
    week[6] = "Saturday";
for (int i = 0; i <= week.Length-1; i++)
listBox1 .Items .Add (week[i]);
}
1 reference
private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Add Week Days";
}
```

C# String Array

Solve the same example using the other declaration

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string[] week = {"Sunday", "Monday","Tuesday","Wednsday"};

    foreach (string Day in week)
        listBox1.Items.Add(Day);
}
1 reference
private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Add Week Days";
    button2.Text = "Clear ";
}
1 reference
private void button2_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
}
```

C# Char Array

How to resize an Array?

```
{
    InitializeComponent();
}
char[] array = { 'A', 'B', 'C', 'D', 'E' };
1 reference
private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i < array.Length; i++)
        listBox1.Items.Add(array[i]);
}
1 reference
private void Form1_Load(object sender, EventArgs e)
{
    button1.Text = "Add Week Days";
    button2.Text = "Clear ";
}
1 reference
private void button2_Click(object sender, EventArgs e)
{
    Array.Resize(ref array, 3);
    listBox1.Items.Clear();
    for (int i = 0; i < array.Length; i++)
    listBox1.Items.Add(array[i]);
}
```

C# Sort Array

Sorting Arrays

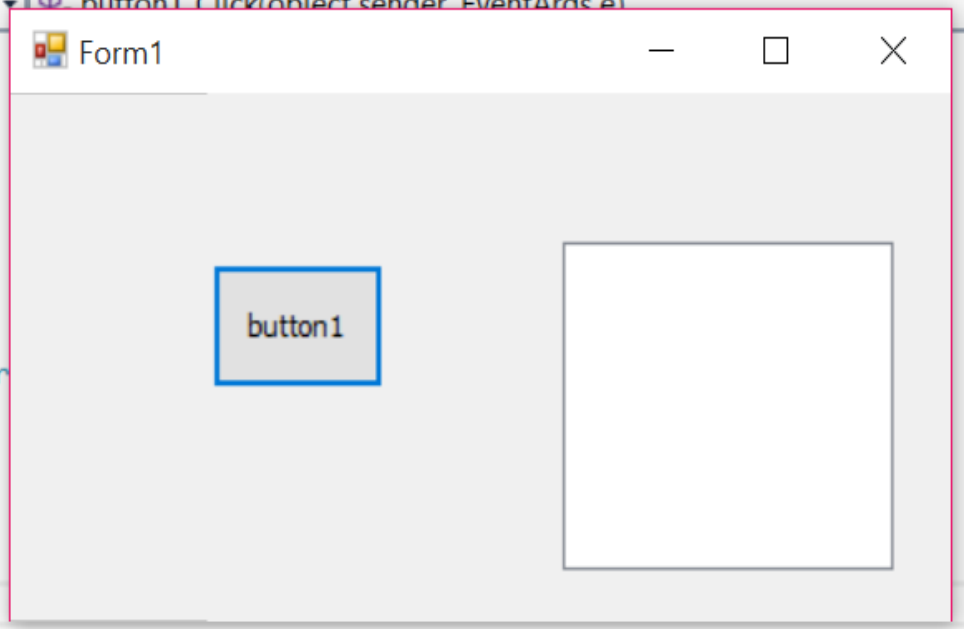
You can sort the arrays in ascending order as well as descending .

We can use Array.Sort method for sorts the elements in a one-dimensional array.

C# Sort char Array

```
rmsApplication12.Form1
```

```
1 reference
public Form1()
{
    InitializeComponent();
}

1 reference
private void button1_Click(object sender, EventAr
{
    char[] letter = { 'Z', 'C', 'A', 'H', 'B' };
    Array.Sort(letter);
    for (int i = 0; i < letter.Length; i++)
        listBox1.Items.Add(letter[i]);
}
```

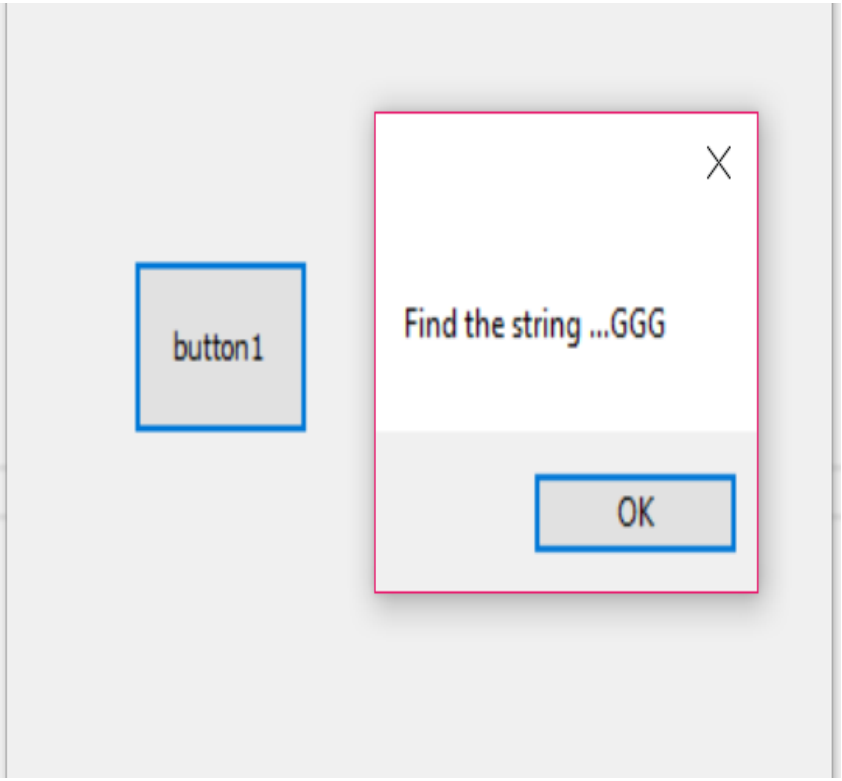


C# Sort char in reverse order

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    char[] letter = { 'Z', 'C', 'A', 'H', 'B' };
    Array.Sort(letter);
    Array.Reverse(letter);
    for (int i = 0; i < letter.Length; i++)
        listBox1.Items.Add(letter[i]);
}
```

How to check if a value exists in an array ?

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string check = "GGG";
    string[] stringArray = { "ABC", "DEF", "GGG", "JKL" };
    foreach (string x in stringArray)
        if (x.Equals(check))
            MessageBox.Show("Find the string ..." + x);



}
```



C# 2D String Array

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    int i = 0;
    int j = 0;
    string[,] A = new string[2, 3];

    A[0, 0] = "First";
    A[0, 1] = "Second";

    A[1, 0] = "Third";
    A[1, 1] = "Fourth";

    for (i = 0; i <= A.GetUpperBound(0); i++)
        for (j = 0; j <= A.GetUpperBound(0); j++)
            MessageBox.Show(A[i, j]);

    }
```

HW

1.Sort the week days example in descending order.

2.Read the two string st1= “C# test program”

st1= “C# program”, then compare between them if equal return massage “String match”

If not equal return massage “String does not match”.

3.St =“for (int count = 0; count < st2.Length - 1; count++)”

using String.split() method, split this string.

How to use C# Directory Class

Directory class in c# exposes methods to create , delete , move etc.

How to create a directory using Directory class in C# ?

In order to create a new directory using Directory class in C# , we can call **CreateDirectory** method directly from Directory class.

Syntax : Directory.CreateDirectory(string DirPath)

DirPath : The name of the new directory

CSharp Code : **Directory.CreateDirectory(“D:\\testDir1");**

How to use C# Directory Class

using System.IO;

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    //create the directory testDir1
    Directory.CreateDirectory("D:\\testDir1");
    MessageBox.Show("testDir1 created ! ");
}
```

How to check a directory exist or not?

Before we creating a directory or folder , we usually check that directory or folder exist or not.

Syntax : bool Directory.Exist(string DirPath)

CSharp Code : Directory.Exists(“D:\\testDir1")

How to check a directory exist or not?

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    if (Directory.Exists("D:\\testDir1"))
    {
        //shows message if testdir1 exist
        MessageBox.Show("Directory 'testDir' Exist ");
    }
    else
    {
        //create the directory testDir1
        Directory.CreateDirectory("D:\\testDir1");
        MessageBox.Show("testDir1 created ! ");
    }
}
```

How to delete a Directory using Directory class in C# ?

When we want to delete a directory we can use the Delete method in the C# Directory class

Syntax : void Directory.Delete(string DirPath)

DirPath : The Directory we want to delete.

CSharp Code : Directory.Delete(“D:\\testDir1");

How to delete a Directory using Directory class in C# ?

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    //delete the directory testDir1
    Directory.Delete("D:\\testDir1");
    MessageBox.Show("testDir1 deleted ");
}
```

How to use C# File Class

File class is using for the File operations in C#. We can create , delete , copy etc. operations do with C# File class.

How to create a File using C#?

In order to create a new File using C# File class , we can call Create method in the File class.

Syntax : FileSttream File.Create(string FilePath)

FilePath : The name of the new File Object

CSharp Code : File.Create(“D:\\testFile.txt");

```
File.Create("D:\\testFile.txt");
```

How to check a File exist or not?

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    if (File.Exists("D:\\testFile.txt"))
    {
        //shows message if testFile exist
        MessageBox.Show("File 'testFile' Exist ");
    }
    else
    {
        //create the file testFile.txt
        File.Create("D:\\testFile.txt");
        MessageBox.Show("File 'testFile' created ");
    }
}
```

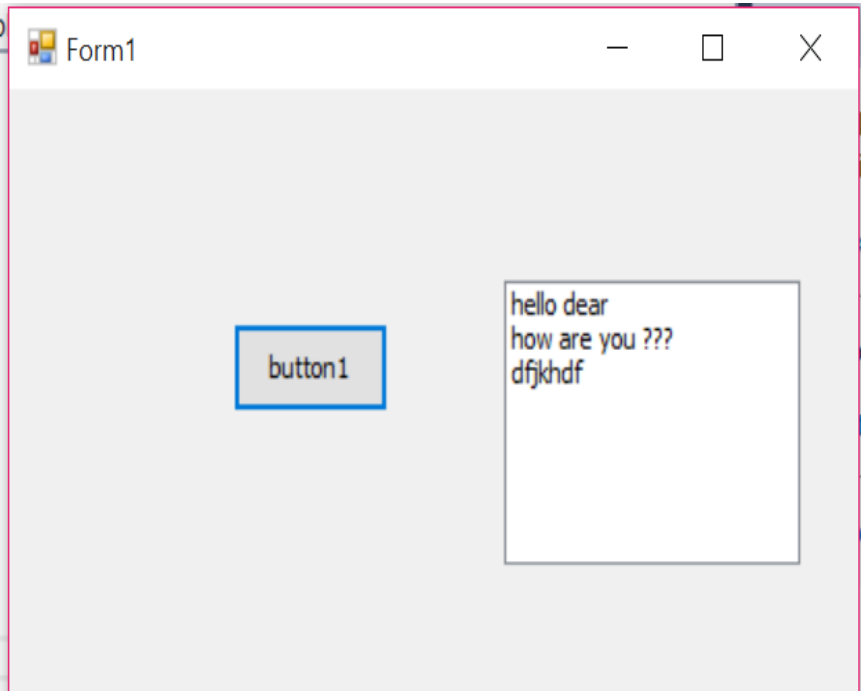
How to use C# Textreader Class

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        string line = null;
        System.IO.TextReader readFile = new StreamReader("D:\\csharp.net-informations.txt");
        while (true)
        {
            line = readFile.ReadLine();
            if (line != null)
            MessageBox.Show(line);
        }

    }
    catch (IOException ex)
    {
        MessageBox.Show(ex.ToString());
    }
```

How to use C# Textreader Class

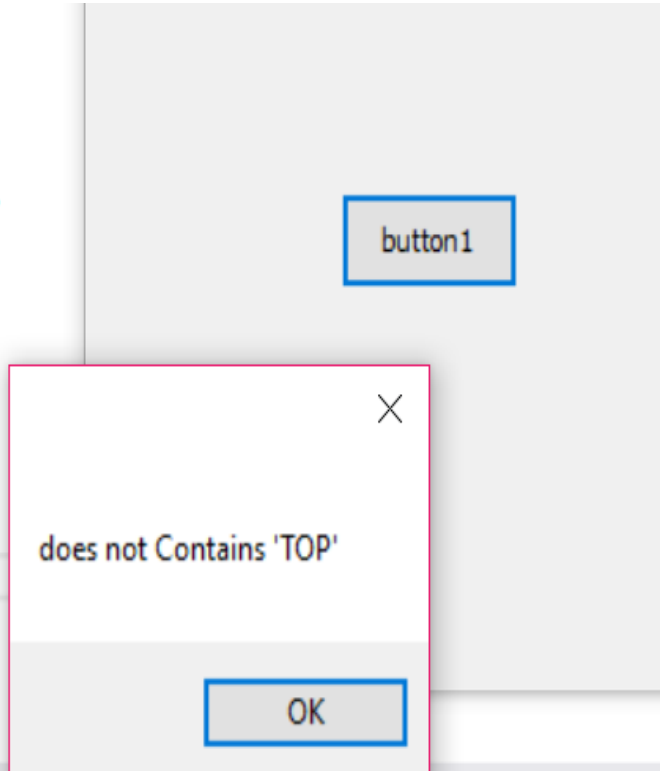
rmsApplication20.Form1 ▾ button1_Click(o

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string line;
    TextReader readFile = new StreamReader("D:\\testFile.txt");
    line = readFile.ReadLine();
    while (line != null)
    {
        listBox1.Items.Add(line);
        line = readFile.ReadLine();
    }


}
```



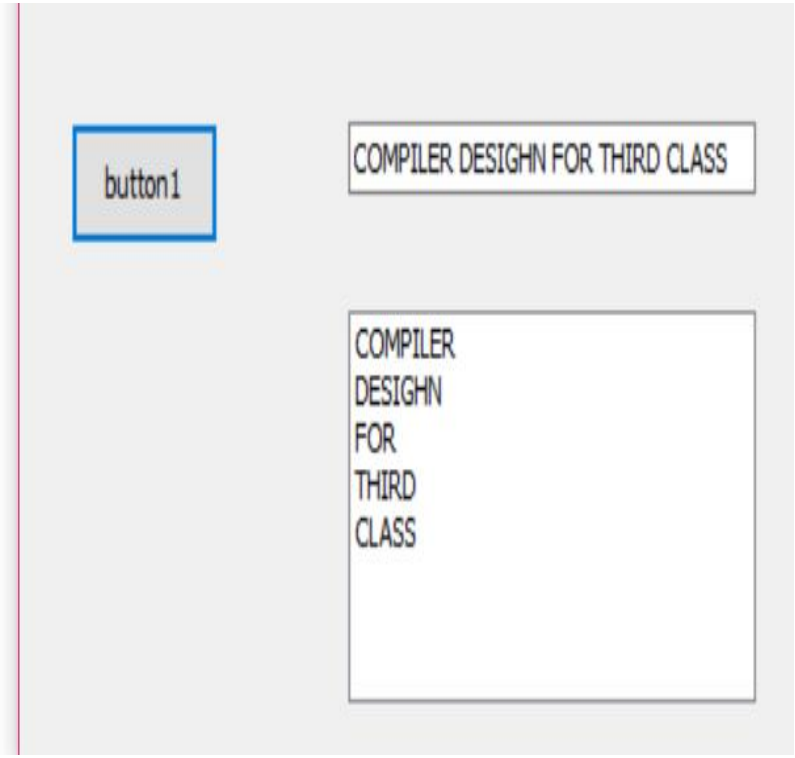
How to use C# Textreader Class

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string line;
    TextReader readFile = new StreamReader("D:\\testFile.txt");
    line = readFile.ReadLine();
    if (line.Contains("TOP") == true)
        MessageBox.Show(" Contains  'TOP' ");
    else
        MessageBox.Show("does not Contains 'TOP'");



}
```



How to use C# Textreader Class

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string s;
    System.IO.TextReader readFile = new StreamReader("D:\\textFile.txt");
    s = readFile.ReadLine();
    textBox1.Text = s;
    char[] ss = { ' ' };
    String[] s2 = s.Split(ss);
    for (int i = 0; i <= s2.Length - 1; i++)
        listBox1.Items.Add(s2[i]);
}
```



How to use C# TextWriter Class

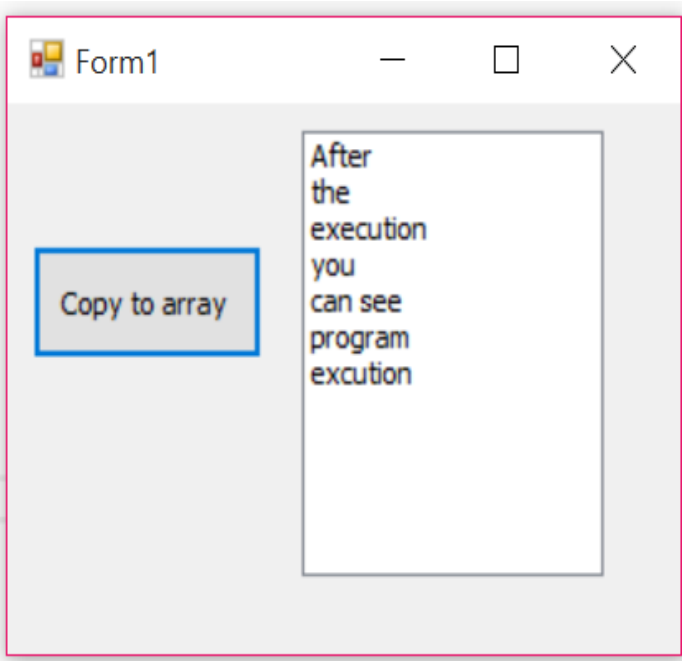
```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    System.IO.TextWriter writeFile = new StreamWriter("D:\\textwriter.txt");
    writeFile.WriteLine("csharp.net-informations.com");
    writeFile.Flush();
    writeFile.Close();
    writeFile = null;
}
```

How to use C# TextWriter Class

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string line = "THIS IS THE THIRD LECTURE";
    System.IO.TextWriter writeFile = new StreamWriter("D:\\textwriter.txt");
    char[] ss = { ' ' };
    String[] s2 = line.Split(ss);
    for (int i = 0; i <= s2.Length - 1; i++)
    writeFile.WriteLine(s2[i]);
    writeFile.Flush();
    writeFile.Close();
    writeFile = null;
}
```

Copy File content to array 1D

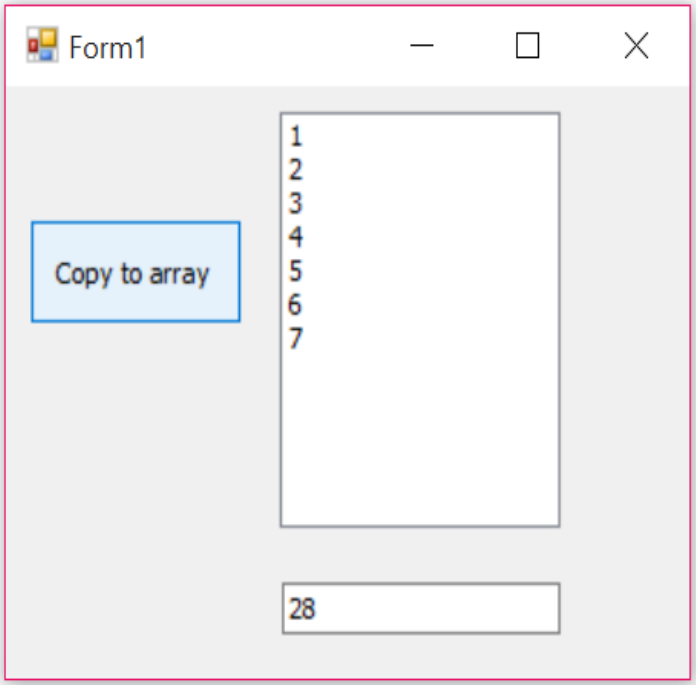
```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    String[] words = new string[7];
    int i = 0;
    string line;
    TextReader readFile = new StreamReader("D:\\Test1.txt");
    line = readFile.ReadLine();
    while (line != null)
    {
        words[i] = line;
        i += 1;
        line = readFile.ReadLine();
    }
   for (int k =0; k < words .Length ; k++)
   {
       listBox1.Items.Add(words [k]);

    }
```



Copy File content to array 1D

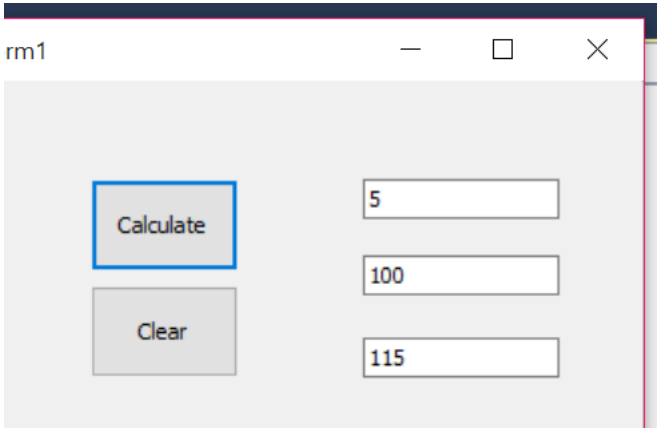
ormsApplication34.Form1 ▾ | button1_Click(object sender, EventArgs e)

```
private void button1_Click(object sender, EventArgs e)
{
    int [] words = new int [7];
    int i = 0;
    string line;
    TextReader readFile = new StreamReader("D:\\Test1.txt");
    line = readFile.ReadLine();
    while (line != null)
    {
        words[i] = int .Parse (line);
        i += 1;
        line = readFile.ReadLine();
    }
    int sum = 0;
    for (int k =0; k < words .Length ; k++)
   {
       listBox1.Items.Add(words [k]);
       sum += (words[k]);
   } textBox1.Text = sum.ToString();
}
```



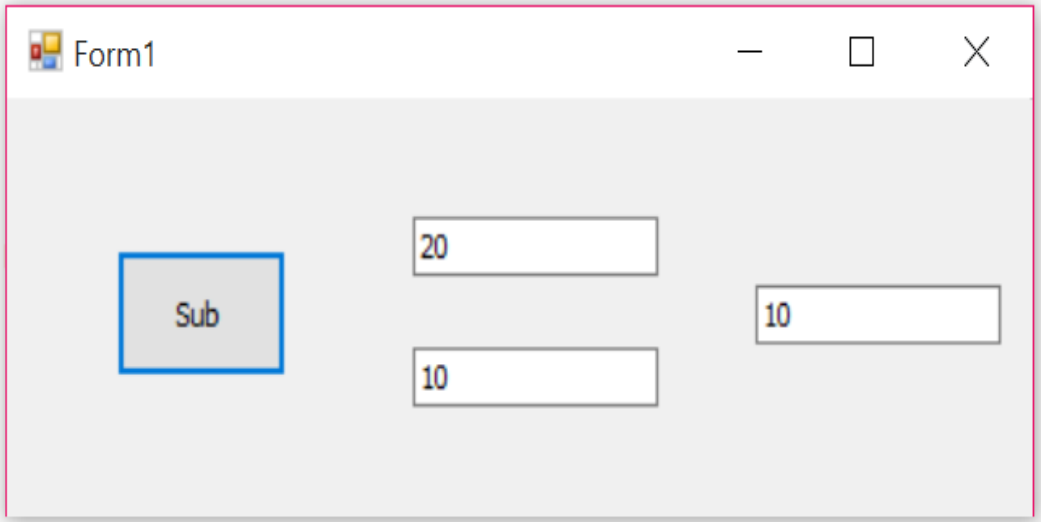
Public

```
1 reference
public int Compute1(int x)
{
    int y = x * 20;
    return y;
}
1 reference
public int Compute2(int x1, int x2, int x3)
{
    int result = x1 + x2 + x3;
    return result;
}
1 reference
private void button1_Click(object sender, EventArgs e)
{
    int c = int.Parse(textBox1.Text);
    int m = Compute1(c);
    textBox2.Text = m.ToString();
    textBox3.Text = Compute2(5, 10, 100).ToString();
}
```



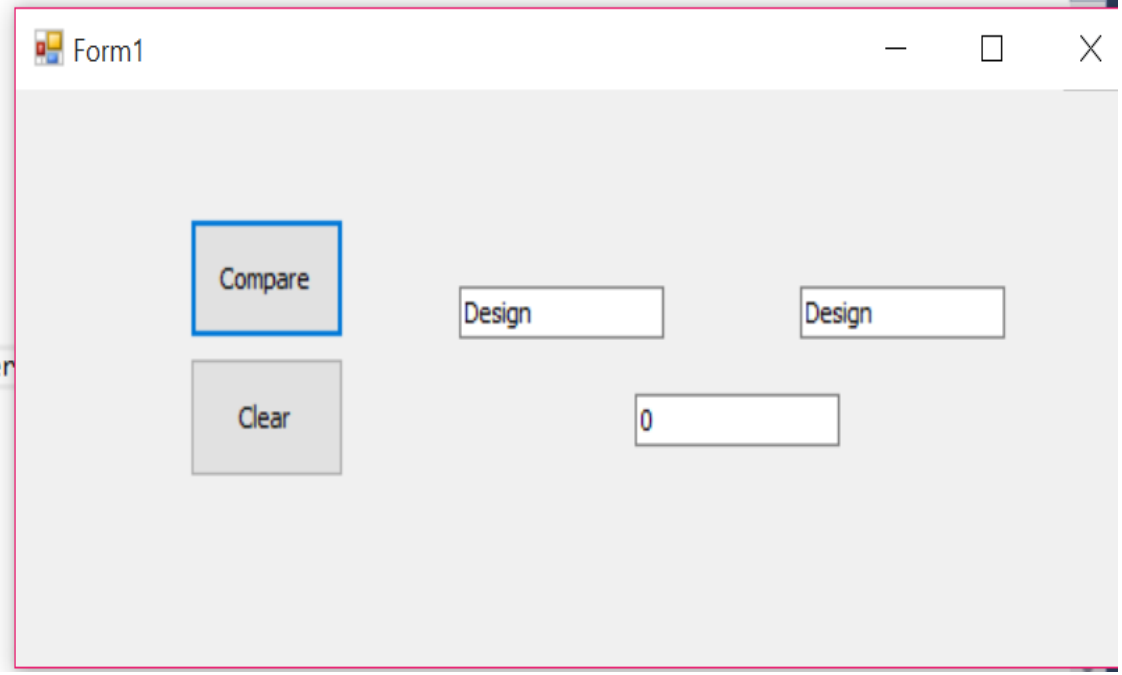
Private

```
1 reference
private  int  Sub(int no1, int no2)
 {
     int result = no1 - no2;
     return result;

 }
 1 reference
 private void button1_Click(object sender,
 {
     int a ,b, z;
     a = int.Parse (textBox1.Text);
     b = int.Parse(textBox2.Text);
     z= Sub(a,b);
     textBox3 .Text =z.ToString();
 }
```



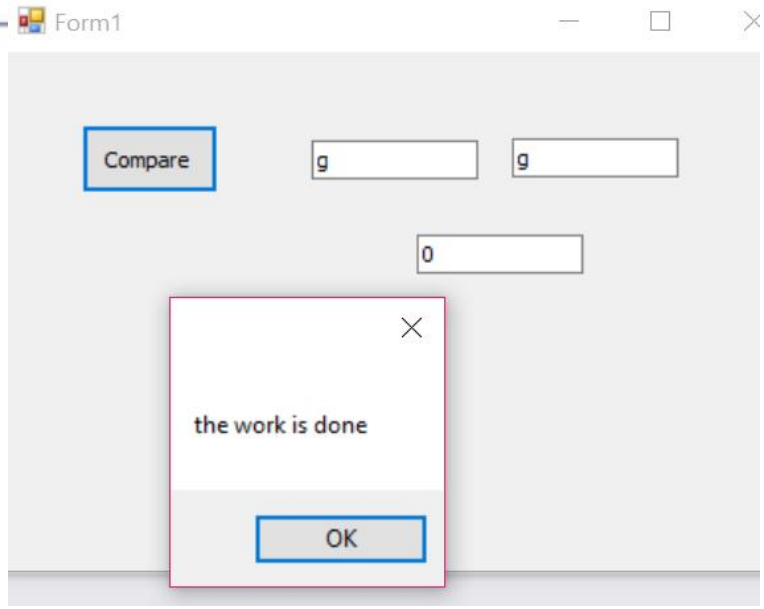
Public

```
vsFormsApplication32.Form1                    button1_Click(object sender, EventArgs e)
        InitializeComponent();
    }
     1 reference
     public int call(string s1, string s2)
    {
        int r = string.Compare(s1, s2);
        return r;
    }
     1 reference
     private void button1_Click(object sender
    {
        string x1 = textBox1.Text;
        string x2 = textBox2.Text;
        int y = call(x1, x2);
        textBox3.Text = y.ToString();
    }
```



Void

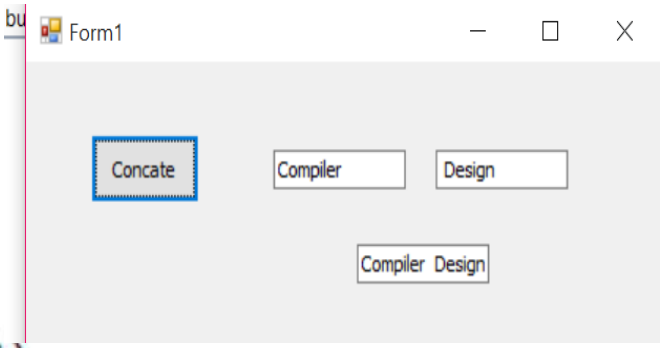
```
}
1 reference
public void Do()
{
    MessageBox.Show("the work is done");
}
1 reference
public int call(string s1, string s2)
{
    int r = string.Compare(s1, s2);
    return r;
}
```



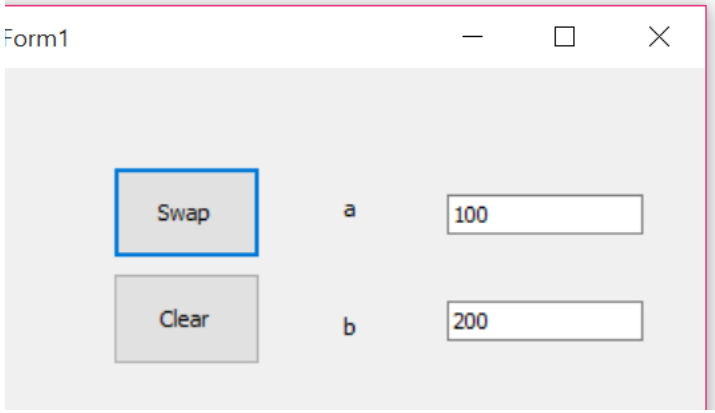

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string x1 = textBox1.Text;
    string x2 = textBox2.Text;
    int y = call(x1, x2);
    textBox3.Text = y.ToString();
    Do();
```

Public

```
1 reference
public Form1()
{
    InitializeComponent();
}
1 reference
public string call1(string s1, string s2)
{
    string r = string.Concat(s1, s2);
    return r;
}
 1 reference
 private void button1_Click(object sender, EventArgs e)
{
    string x1 = textBox1.Text;
    string x2 = textBox2.Text;
   textBox3.Text = call1(x1, x2);
   //textBox3.Text = call1(textBox1.Text, textBox2.Text);
}
```



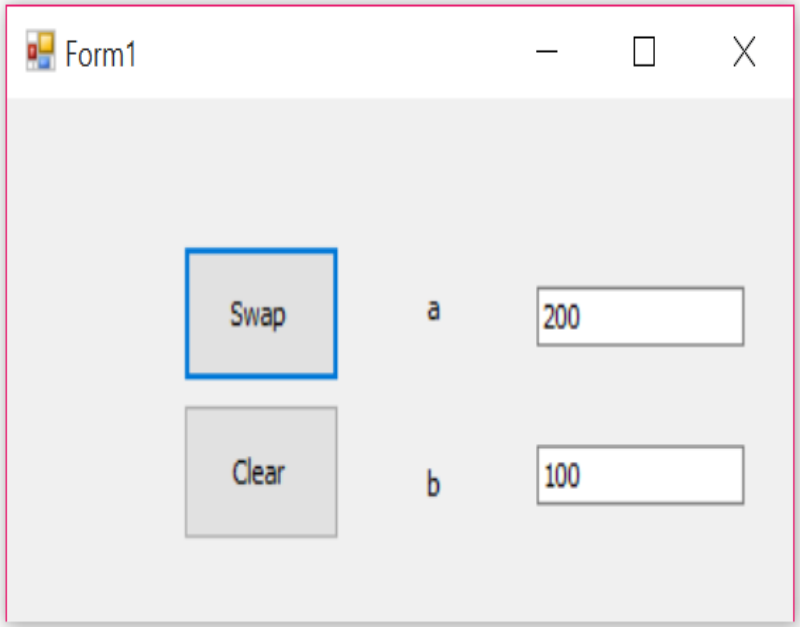
PASSING PARAMETERS BYVAL



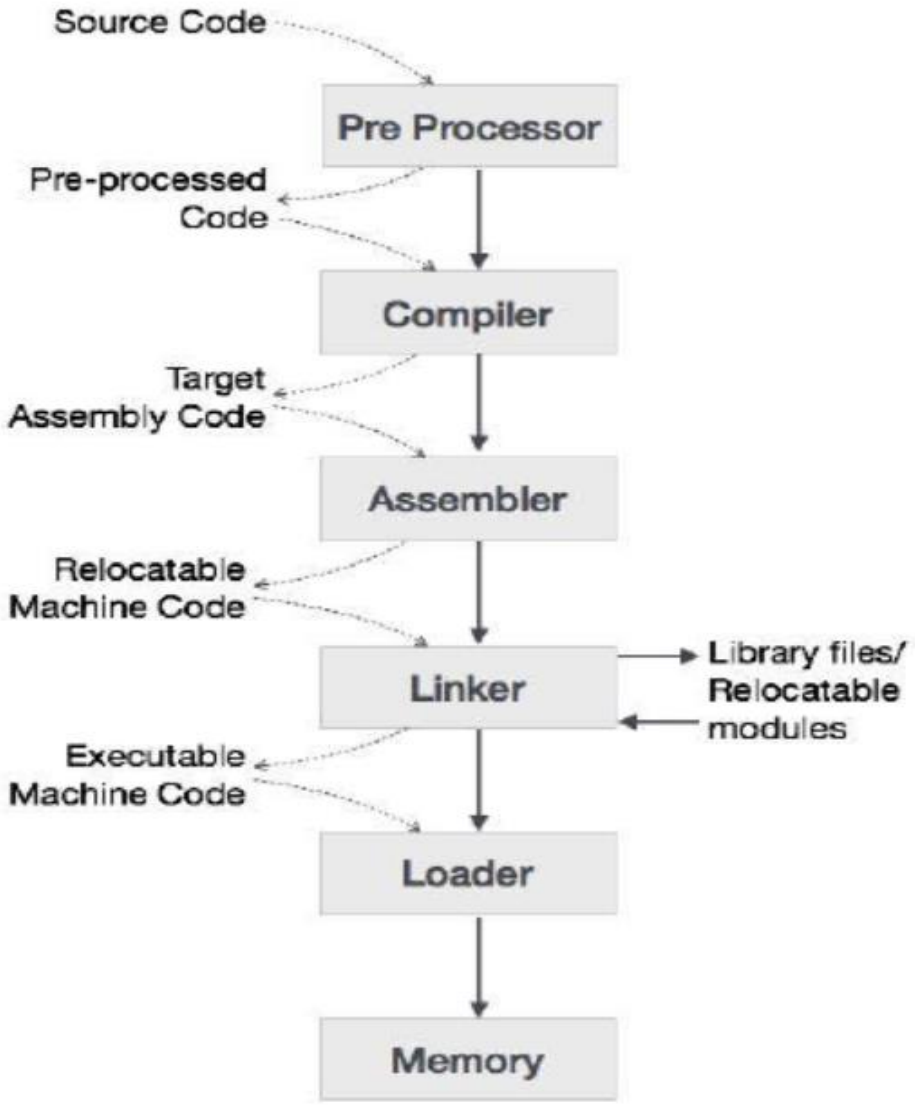

```
1 reference
public void swap(int x, int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
1 reference
private void button1_Click(object sender, EventArgs e)
{
    int a = 100;
    int b = 200;
    swap(a, b);
    textBox1.Text = a.ToString();
    textBox2.Text = b.ToString();
}
```

PASSING PARAMETERS BYVAL BYREF

```
1 reference
public void swap(ref int x, ref int y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}
1 reference
private void button1_Click(object sender, EventArgs e)
{
    int a = 100;
    int b = 200;
    swap(ref a, ref b);
    textBox1.Text = a.ToString();
    textBox2.Text = b.ToString();
}
```

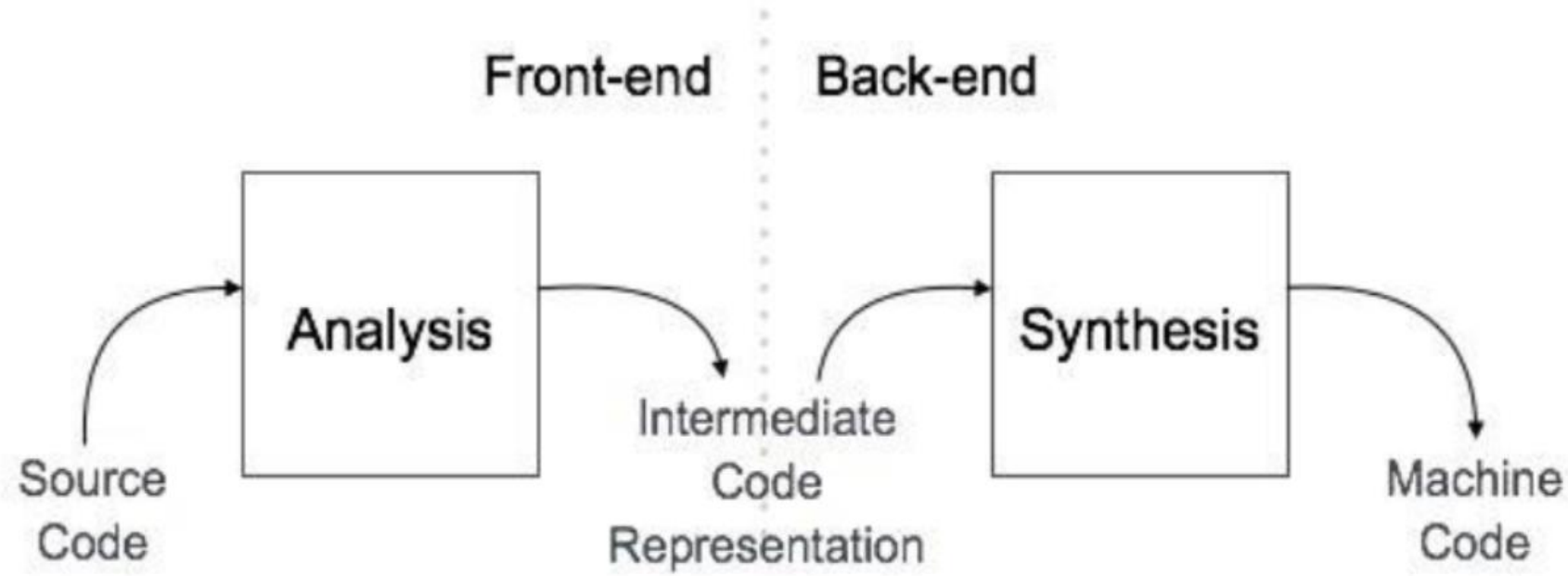


Language Processing System

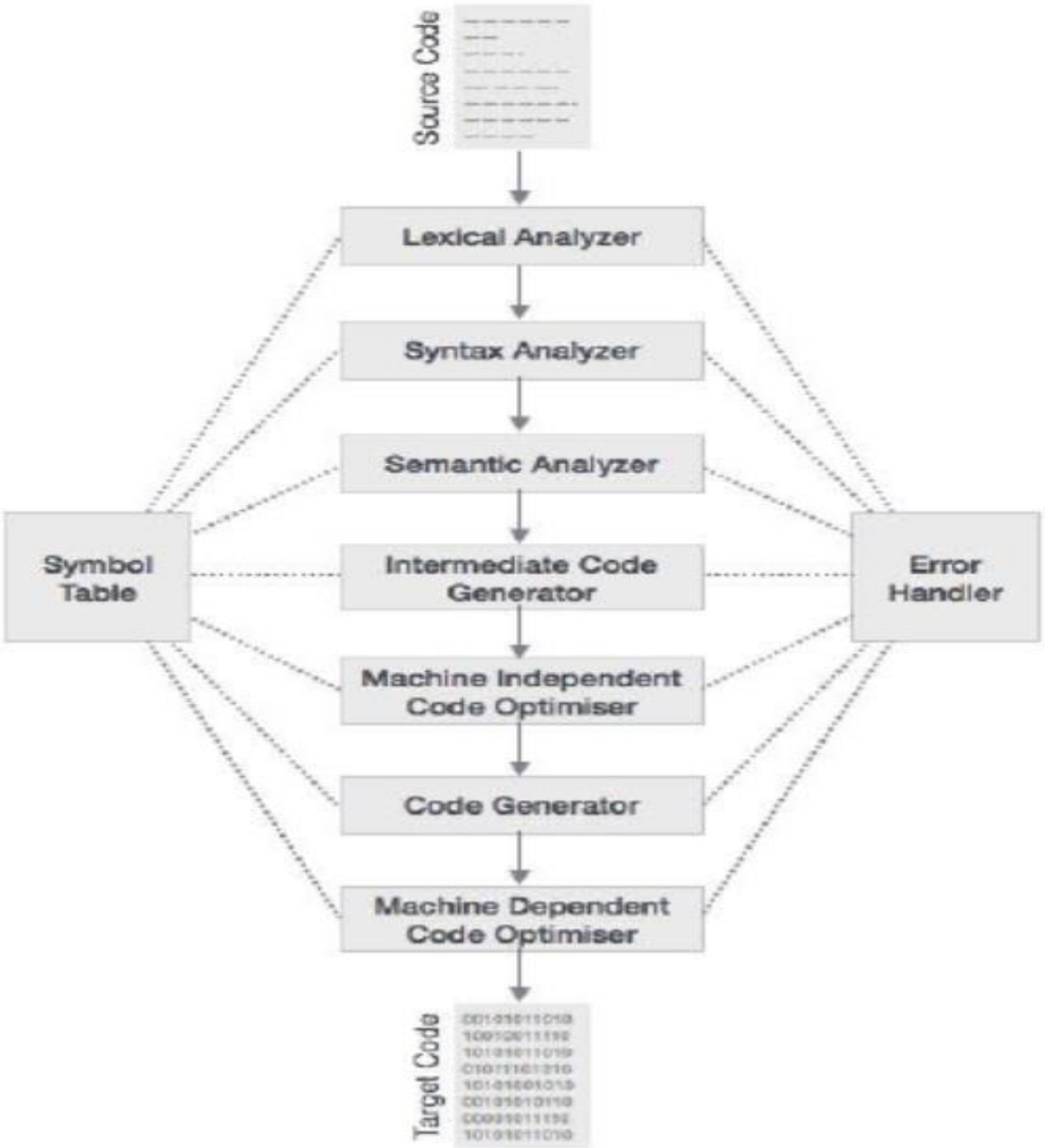


COMPILER ARCHITECTURE

Analysis Phase , Synthesis Phase

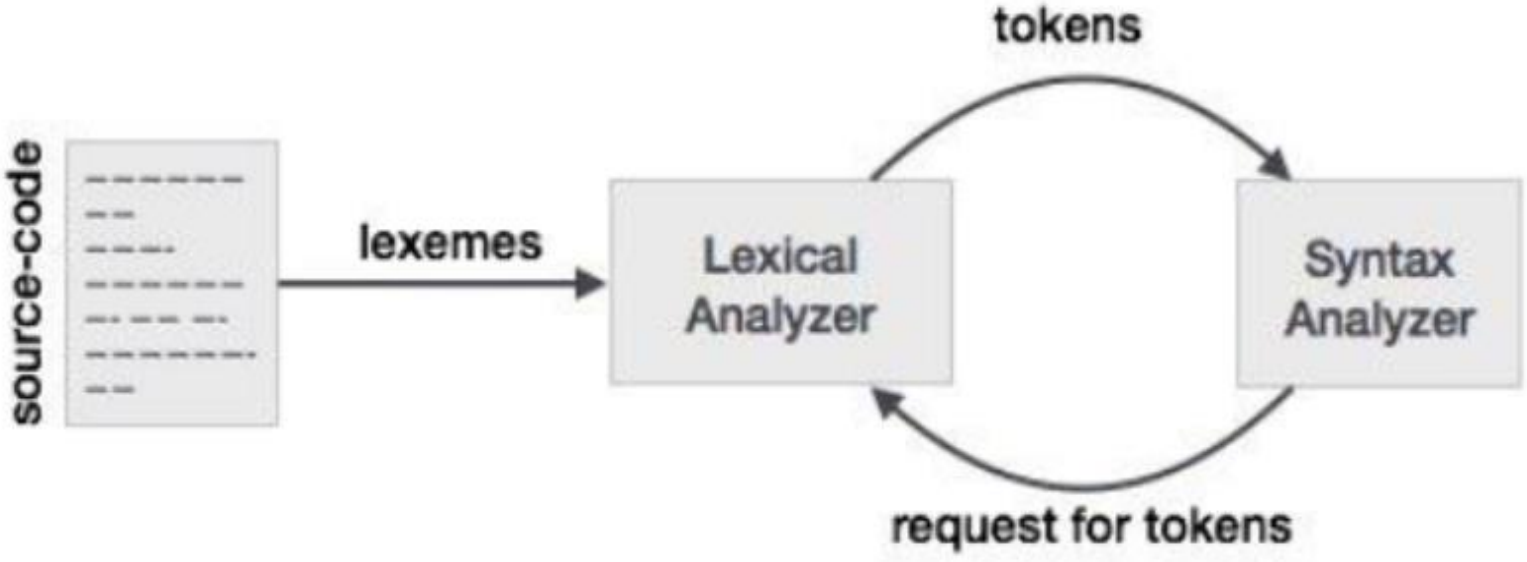


PHASES OF COMPILER



LEXICAL ANALYSIS

The lexical analyzer **breaks these syntaxes** into a **series of tokens**, by removing any whitespace or comments in the source code



LEXICAL ANALYSIS

Lexemes are said to be a sequence of characters (alphanumeric) in a token. There are some predefined rules for every lexeme to be identified as a valid token. These rules are defined by grammar rules, by means of a pattern. A pattern explains what can be a token, and these patterns are defined by means of regular expressions.

int *value* = 100;

REGULAR EXPRESSIONS

Representing valid tokens of a language in regular expression

If x is a regular expression, then:

- x* means zero or more occurrence of x.

i.e., it can generate { e, x, xx, xxx, xxxx, … }

- x+ means one or more occurrence of x.

i.e., it can generate { x, xx, xxx, xxxx … } or x.x*

- x? means at most one occurrence of x

i.e., it can generate either {x} or {e}.

[a-z] is all lower-case alphabets of English language.

[A-Z] is all upper-case alphabets of English language.

[0-9] is all natural digits used in mathematics.

Representing occurrence of symbols using regular expressions

letter = [a – z] or [A – Z]
digit = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 or [0-9]
sign = [+ | -]

Representing language tokens using regular expressions

Decimal = (sign)?(digit)+
Identifier = (letter)(letter | digit)*

The only problem left with the lexical analyzer is how to verify the validity of a regular expression used in specifying the patterns of keywords of a language. A well-accepted solution is to use finite automata for verification.

FINITE AUTOMATA

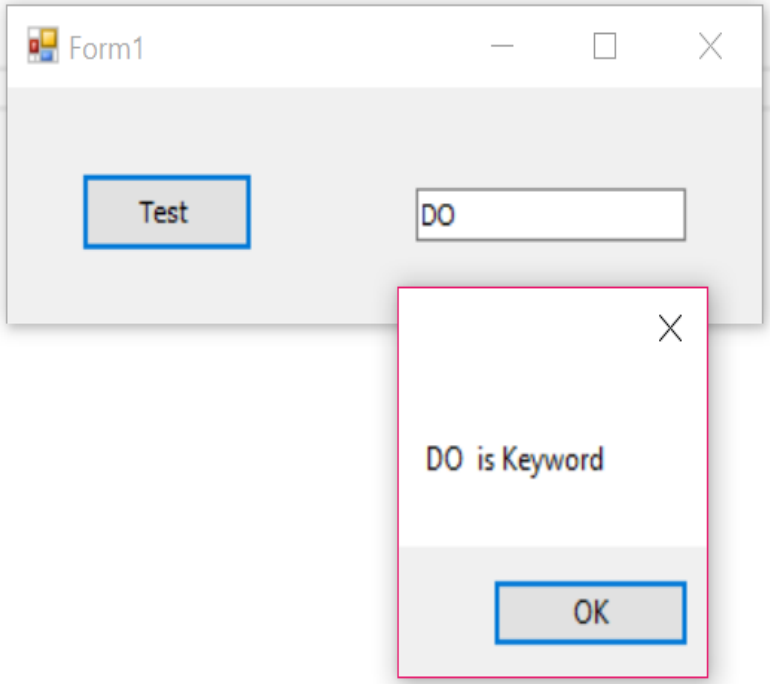
Finite automata is a recognizer for regular expressions

The mathematical model of finite automata consists of:

- Finite set of states (Q)
- Finite set of input symbols (Σ)
- One Start state (q0)
- Set of final states (qf)
- Transition function (δ)

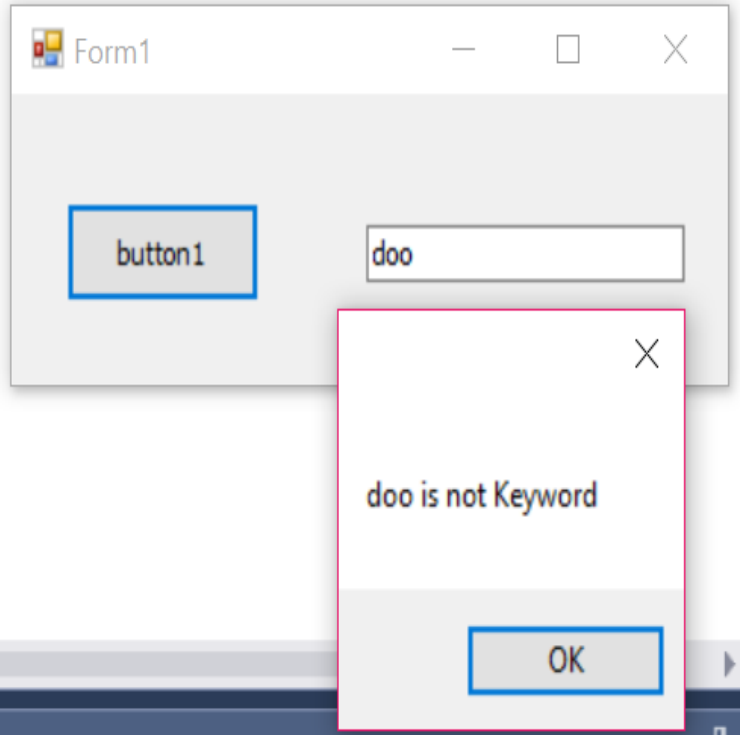
Token - Keywords

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string word = textBox1.Text;
    string[] keywords = { "for", "if", "else", "do", "switch", "null", "case", "end", "while", "int" , "string"
    bool x = false;
    for (int j = 0; j < keywords.Length; j++)
        if (keywords[j] == word.ToLower ())
        {
            MessageBox.Show(word +"  is Keyword ");
            x = true;
        }
    if (!x)
        MessageBox.Show( word +" is Not Keyword ");
}
```



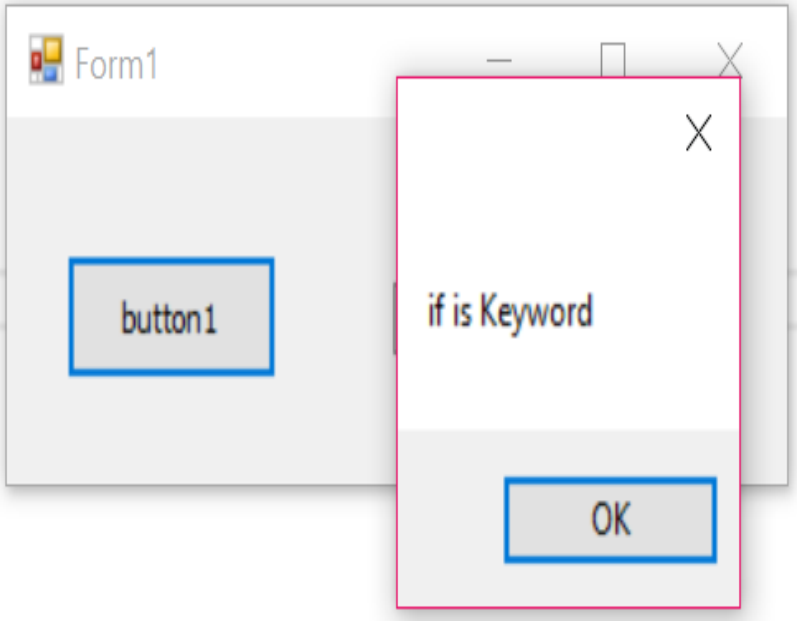
Token - Keywords

```
private void button1_Click(object sender, EventArgs e)
{
    string[] Keyword = new string[8] { "for", "if", "do", "int", "else", "while", "int", "string" };
    string value = textBox1.Text;
    if (Keyword.Contains(value.ToLower()))
        MessageBox.Show(value.ToString() + " is Keyword");
    else
        MessageBox.Show(value.ToString() + " is not Keyword");
}
```



Token

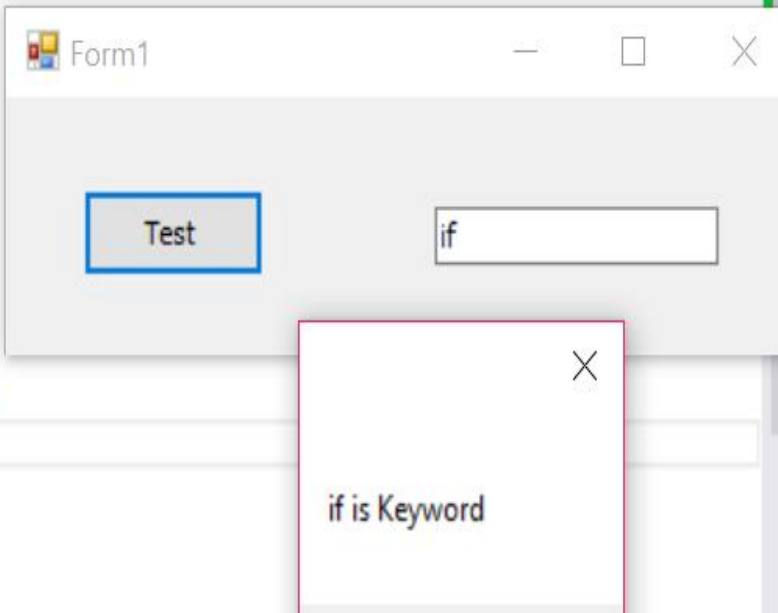
```
private void button1_Click(object sender, EventArgs e)
{
    string[] Keyword = new string[8] { "for", "if", "do", "int", "else", "while", "int", "string" };
    string [] value = new string[2]{"5","if"};
     foreach (string v in value )
    if (Keyword.Contains(v.ToLower()))
        MessageBox.Show(v.ToString() + " is Keyword");
    else
        MessageBox.Show(v.ToString() + " is not Keyword");
}
```



Token

keywords , operation, logic operation, punctuation

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string[] operation = new string[4] { "+", "-", "/", "*" };
    string[] Keyword = new string[8] { "for", "if", "do", "int", "else", "while", "int", "string" };
    string[] logicoperation = new string[6] { ">", "<", "==", "<=", ">=", "!=" };
    string[] puncuation = new string[6] { "{", "}", "(", ")", ";", "," };
    string value = textBox1.Text;
    if (operation.Contains(value))
        MessageBox.Show(value.ToString() + " is operation");
    else if (Keyword.Contains(value.ToLower()))
        MessageBox.Show(value.ToString() + " is Keyword");
    else if (puncuation.Contains(value))
        MessageBox.Show(value.ToString() + " is puncuation");
    else if (logicoperation.Contains(value))
        MessageBox.Show(value.ToString() + " is logic operation");

}
```

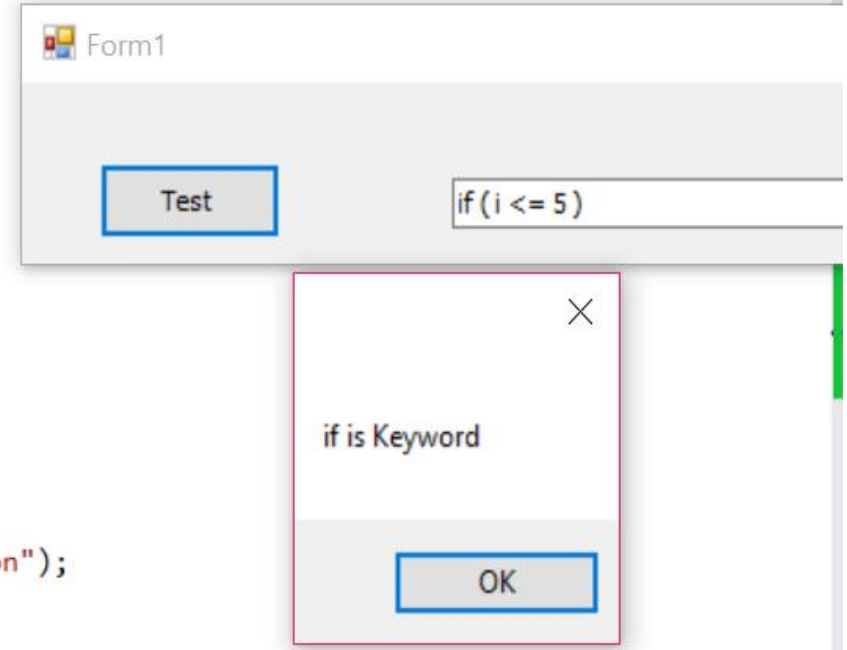


Token

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string[] operation = new string[4] { "+", "-", "/", "*" };
    string[] Keyword = new string[8] { "for", "if", "do", "int", "else" , "while", "int", "string"};
    string[] logicoperation = new string[6] { ">", "<", "==", "<=", ">=", "!=" };
    string[] puncuation = new string[6] { "{", "}", "(", ")", ";", "," };
    string [] value = new string [3] {"(","if", "mam"};
    foreach (string v in value)
    {
        if (operation.Contains(v))
            MessageBox.Show(v.ToString() + " is operation");
        else if (Keyword.Contains(v.ToLower()))
            MessageBox.Show(v.ToString() + " is Keyword");
        else if (puncuation.Contains(v))
            MessageBox.Show(v.ToString() + " is puncuation");
        else if (logicoperation.Contains(v))
            MessageBox.Show(v.ToString() + " is logic operation");
        else MessageBox.Show(v.ToString() + " is wrong");
    }
}
```

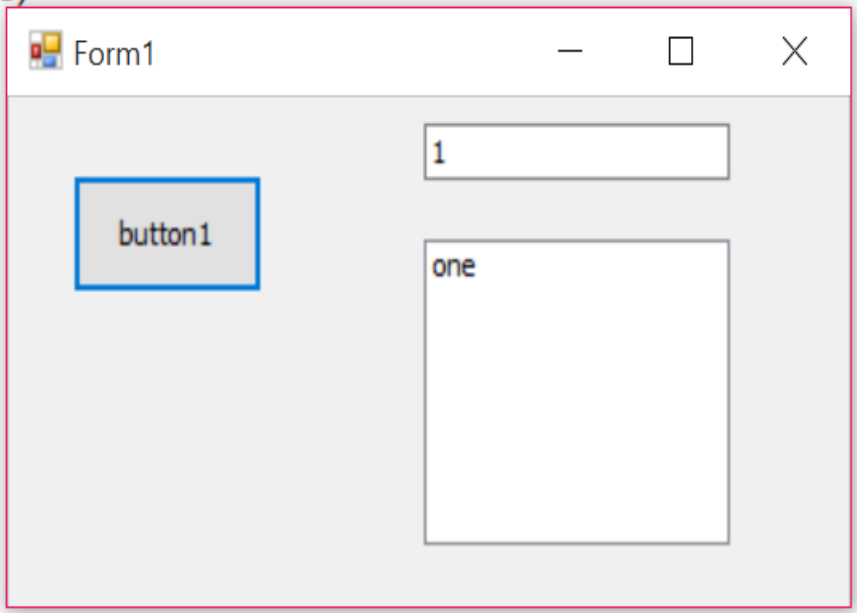
Token -from string

```
private void button1_Click(object sender, EventArgs e)
{
    string[] operation = new string[4] { "+", "-", "/", "*" };
    string[] Keyword = new string[8] { "for", "if", "do", "int", "else" , "while", "int", "string"};
    string[] logicoperation = new string[6] { ">", "<", "==", "<=", ">=", "!=" };
    string[] puncuation = new string[6] { "{", "}", "(", ")", ";", "," };
    string[] no = new string[10] { "0", "1", "2", "3", "4", "5", "6", "7", "8", "9" };
    string va= textBox1 .Text;
    char[] delemeters = { ' ' };
     string [] value = null;
    value = va.Split(delemeters);
    foreach (string v in value)
    {
        if (operation.Contains(v))
            MessageBox.Show(v.ToString() + " is operation");
        else if (Keyword.Contains(v.ToLower()))
            MessageBox.Show(v.ToString() + " is Keyword");
        else if (puncuation.Contains(v))
            MessageBox.Show(v.ToString() + " is puncuation");
        else if (logicoperation.Contains(v))
            MessageBox.Show(v.ToString() + " is logic operation");
        else if (no.Contains(v))
            MessageBox.Show(v.ToString() + " is no");
        else MessageBox.Show(v.ToString() + " is wrong");
    }
```

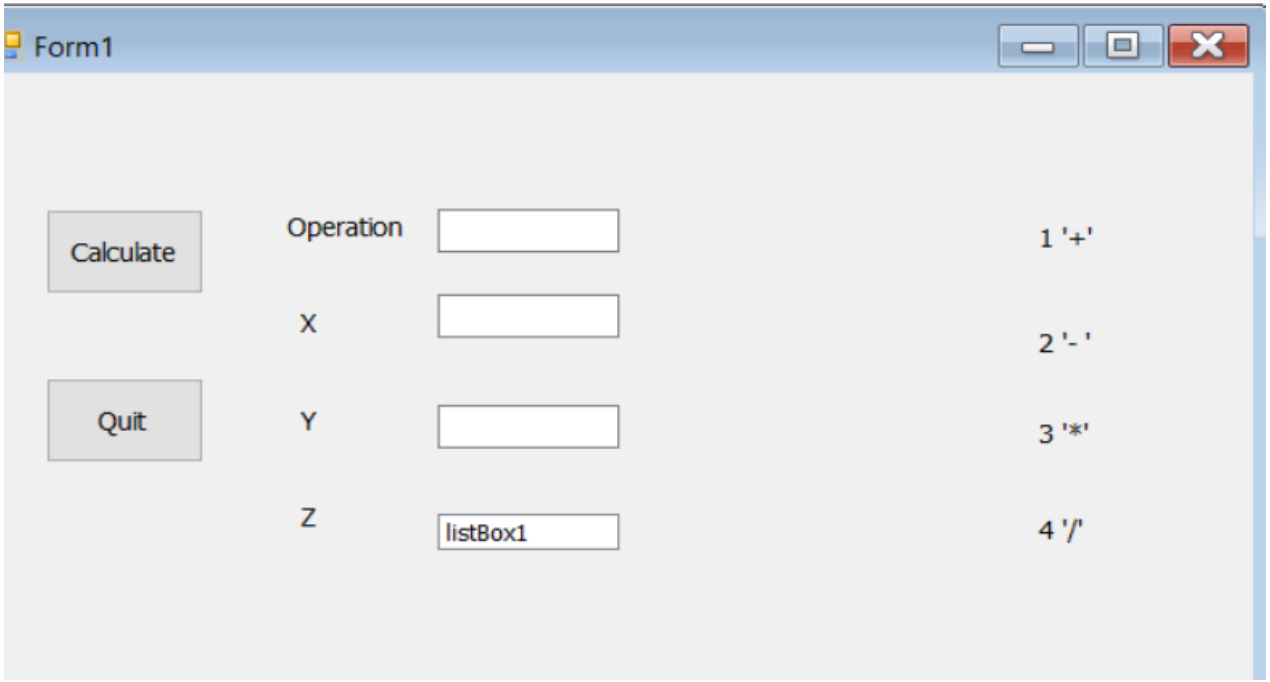


Switch Syntax

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    int  Value;
    Value = int.Parse (textBox1.Text);
    switch (Value)
    {
        case 1:
            listBox1.Items.Add("one");
            break;
        case 2:
            listBox1.Items.Add("two");
            break;
        case 3:
            listBox1.Items.Add("three");
            break;
        default:
            listBox1.Items.Add(" out range 1-3");
            break;
    }
```

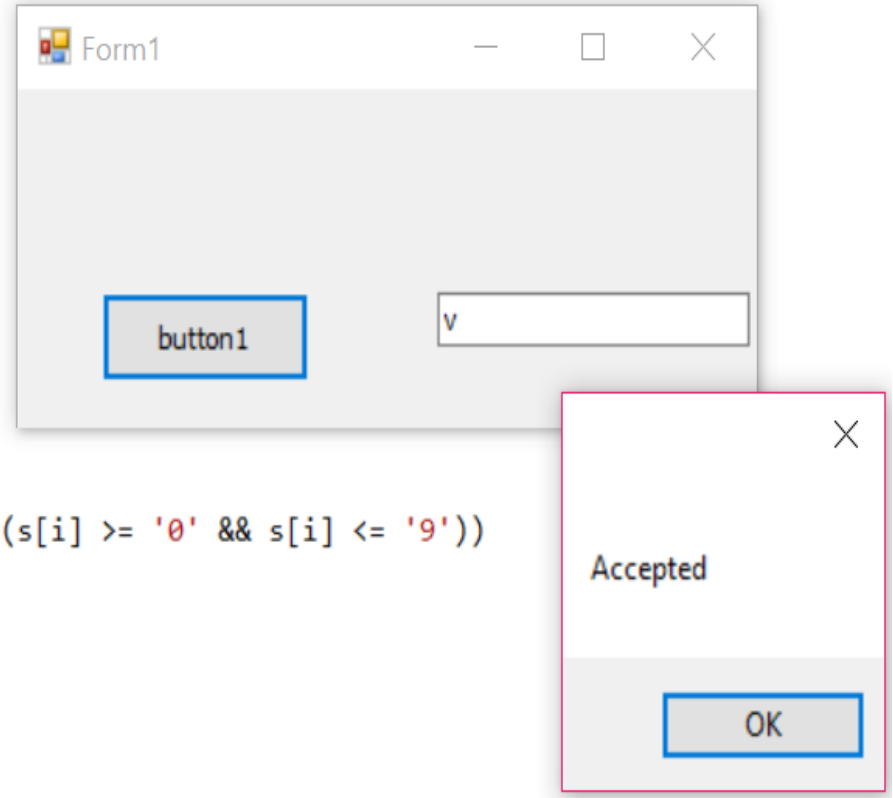


Switch Syntax



Identifier

```
{
    string s = textBox1.Text;
    int state = 0;
    for (int i = 0; i < s.Length; i++)
    {
        switch (state)
        {
            case 0: if (s[i] >= 'a' && s[i] <= 'z')
                        state = 1;
                    else
                        state = 200;
                    break;
            case 1: if ((s[i] >= 'a' && s[i] <= 'z') || (s[i] >= '0' && s[i] <= '9'))
                        state = 1;
                    else
                        state = 200;
                    break;
        }

    } if (state == 1)
        MessageBox.Show("Accepted ");
    else
        MessageBox.Show("Rejected ");
}
```

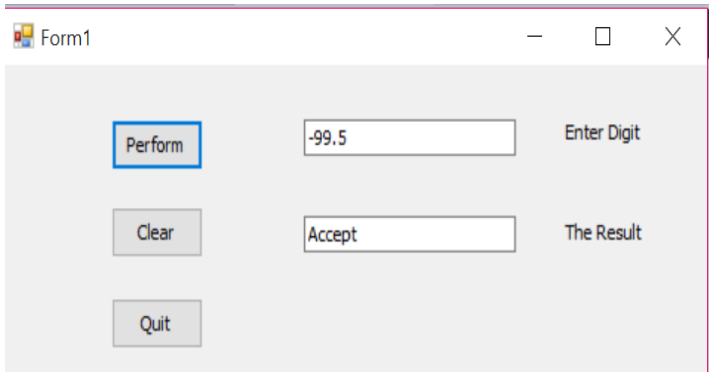


Identifier

```
{
    string s = textBox1.Text;
    int state = 0;
   foreach (char c in s)
    {
        switch (state)
        {
            case 0: if (c >= 'a' && c <= 'z')
                    state = 1;
                else
                    state = 200;
                break;
            case 1: if ((c >= 'a' && c <= 'z') || (c >= '0' && c <= '9'))
                    state = 1;
                else
                    state = 200;
                break;
        }

    } if (state == 1)
        MessageBox.Show("Accepted ");
    else
        MessageBox.Show("Rejected ");
}
```

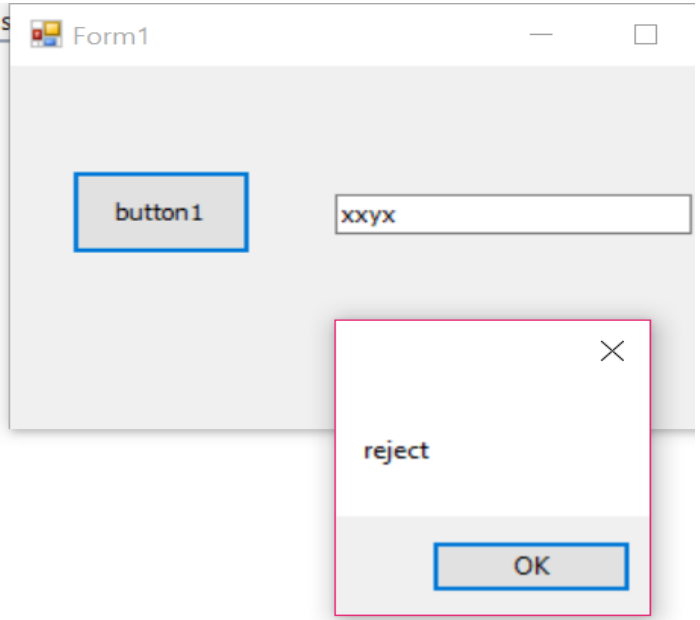
Constant

```
1 reference
private void button1_Click(object sender, EventArgs e)
{ // (+/-)*D+.D+
    string line = textBox1.Text ;
    int state = 2;
    for (int i = 0; i < line.Length; i++)
    {
        switch (state)
        {
            case 2: if (line[i] == '-' || line[i] == '+')
                    state = 3;
            else
               if (line[i] >= '0' && line[i] <= '9')
                    state = 3;
                else
                    state = 300;
                break;
            case 3: if (line[i] >= '0' && line[i] <= '9')
                    state = 3;
                else if (line[i] == '.')
                    state = 4;
                else
                    state = 300;
                break;
            case 4: if (line[i] >= '0' && line[i] <= '9')
                    state = 5;
                else
                    state = 200;
                break;
          case 5: if (line[i] >= '0' && line[i] <= '9')
                    state = 5;
                else
                    state = 200;
                  break;
        }
}
    if (state  == 3 || state == 5)
        textBox3 .Text ="Accept";
    else
        textBox3.Text = "Reject";
```



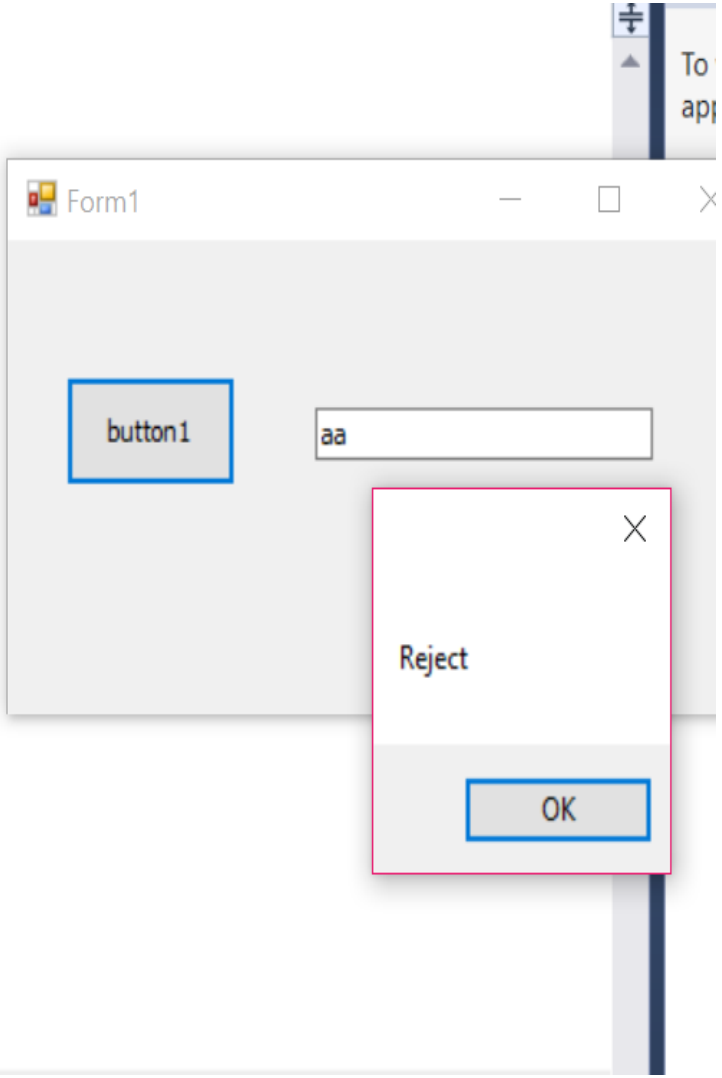
rmsApplication38.Form1 ▾ button1_Click(object s

```
private void button1_Click(object sender, EventArgs e)
{
    string state = "0";
    string st = textBox1 .Text ;
    for (int i = 0; i < st.Length; i++)
    {
        switch (state)
        {
            case "0":
                if (st[i] == 'x')
                    state = "1";
                else
                    state = "4";
                break;
            case "1":
                if (st[i] == 'x')
                    state = "1";
                else
                if (st[i] == 'y')
                    state = "2";
                else
                    state = "4";
                break;

            case "2":
                 if (st[i] == 'y')
                     state = "2";
                 else
                 if (st[i] == 'x')
                     state = "1";
                 else
                     state = "4";
                 break;
             }
    }
    if (state == "2")
        MessageBox .Show ("accept");
    else
        MessageBox.Show("reject");
```



Example

```
private void button1_Click(object sender, EventArgs e)
{
    bool x = true;
    string st = textBox1.Text;
    if (st[0] == 'a' && st.Length == 1)
        x = true;
    else if (st[0] == 'b' || st[0] == 'b' && st[st.Length - 1] == 'c')
    {
        for (int i = 1; i < st.Length - 2; i++)
            if (st[1] == 'b' && st[i] == 'c')
                x = true;
            else
            {
                x = false;
                break;
            }
    }
    else x = false;

    if (x == true)
        MessageBox.Show("Accept");
    else
        MessageBox.Show("Reject");
}
```



Example

E ⟶ EAE E is letter

A ⟶ +|-|*|/

Example

```
private void button1_Click(object sender, EventArgs e)
{
    string x = textBox1.Text;
    int state = 0;
    for (int i = 0; i < x.Length; i++)
    {
        switch (state)
        {
            case 0:
                if (x[i] >= 'a' && x[i] <= 'z')
                    state = 1;
                else
                    state = 55;
                break;
            case 1:
                if (x[i] == '+' || x[i] == '-' || x[i] == '/' || x[i] == '*')
                    state = 2;
                else
                    state = 55;
                break;
            case 2:
                if (x[i] >= 'a' && x[i] <= 'z')
                    state = 3;
                else
                   state = 55;
               break;
           case 3:
                if (x[i] >= 'a' && x[i] <= 'z')
                    state = 3;
                else
                    if (x[i] == '+' || x[i] == '-' || x[i] == '/' || x[i] == '*')
                        state = 2;
                    else
                        state = 55;
                break;
        }
    }
    if (state == 3)
        MessageBox.Show("Accepted");
    else
        MessageBox.Show("Rejected");
```