



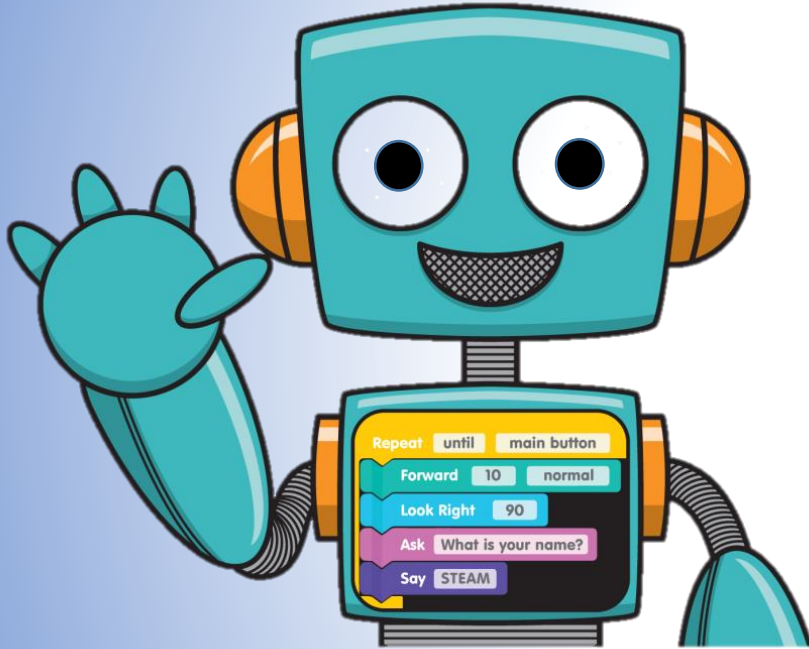
درس برنامه نویسی پیشرفته ۱

مبانی کامپیوتر و برنامه نویسی مقدماتی

مدرس: مسعود بایمانی

فصل چہارم

برنامہ نویسی بہ زبان سی شارپ



ساختار برنامه در زبان سی شارپ



- ساختار اصلی یک برنامه در زبان C# به صورت زیر می باشد. همانگونه که مشاهده می نمایید این ساختار شامل ۶ بخش اصلی می باشد.

```
using System;
using System.Text;

namespace MyAppName
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Welcome to C# Programming!");
        }
    }
}
```

فضاهای نام های استاندارد

فضای نام

کلاس برنامه

تابع اصلی

بدنه تابع

عبارت خاتمه دهنده دستورات

دستور Using



- کامپایلر سی شارپ به منظور اجرای برنامه ها نیاز به فراخوانی برخی از کلاس ها، ساختارها و توابع از پیش تعریف شده دارد تا بتواند اعمال مورد نظر خود را انجام دهد و یک برنامه را کامپایل نماید.
- به فایل های در برگیرنده این کلاس ها، ساختارها و توابع فایل های سرآیند (Header File) گفته می شود.
- از این دستور به منظور رجوع به فضاهای نام و توابع استاندارد زبان استفاده می شود و نحوه کار آن در بیانش نهفته است: «از فضای نام system استفاده کن»

```
using System;
```

- اگر از این دستور در ابتدای برنامه استفاده نشود به منظور ارجاع به توابع هر فضای نام می بایست از نام آن فضای نام به همراه عملگر . قبل از نام توابع استفاده نمود.
- `System.Console.WriteLine("Welcome to C# Programming!");`

فضای نام



- فضاهای نام، محل های برای ذخیره و دسته بندی کلاس ها، ساختارها و توابع می باشند.
- فضاهای نامی دارای خصوصیات زیر هستند:
- فضاهای نام، پروژه هایی با کدها بسیار زیاد را سازماندهی می کنند.
- فضاهای نام با استفاده از عملگر . از یکدیگر تفکیک می شوند.

```
using System.Text;
```

- فضای نام global ، فضای نام root نام دارد و این فضای نام همیشه به فضای نام System.NET اشاره دارد.

```
namespace MyAppName
{
    class Program
    {
        global::System.Console.WriteLine("Hello!");
    }
}
```

تابع Main



- تابع Main، مهمترین جزء از اجزای یک برنامه در زبان سی شارپ است و هیچ برنامه ای وجود ندارد که فاقد تابع Main باشد. هر آنچه که در بدنه این تابع نوشته شود پس از کامپایل اجرا می گردد.

```
static void Main(string[] args)
{
    Console.WriteLine("Welcome to C# Programming!");
}
```

$$y = f(x) = x^2$$

پارامترهای ورودی
(Arguments)



- تمامی دستورات در زبان سی شارپ به ; ختم می شوند.

- کامپایلر سی شارپ در زمان کامپایل دستورات با رسیدن به علامت ; متوجه رسیدن به انتهای دستور میشود.

متغیرهای در زبان C#



• تعریف متغیر در الگوریتم و فلوجارت:

متغیر ظرفی است از جنس حافظه دارای نام که در هر لحظه تنها یک مقدار را در خود جا میدهد.

• تعریف متغیرها در زبان های برنامه نویسی:

متغیرها نام هایی برای خانه های حافظه اند که داده ها در آنها قرار میگیرند و محتویات آنها در طول اجرای برنامه ممکن است بارها و بارها تغییر کند.

• همانطور که می دانیم برای به مراجعه به متغیرها از نامشان استفاده می گردد؛ لذا زبان برنامه نویسی دارای قواعدی برای نام گذاری متغیرها می باشد.

قوانین نام گذاری متغیرها در زبان C#



- ۱- در نام گذاری متغیرها، از حروف الفبایی، ارقام و زیرخط (Underscore) استفاده می شود و می بایست نام متغیر حتما با یکی از حروف الفبایی (a-z or A-Z) یا زیرخط شروع شود.
- ۲- سی شارپ در نام گذاری متغیرها، بین حروف کوچک و بزرگ تفاوت قائل می شود. به طور مثال Xy با XY و xY و xy متفاوت است.
- ۳- استفاده از کاراکترهایی نظیر *، +، /، ؛، @، # و... در نامگذاری متغیرها ممنوع می باشد.
- ۴- استفاده از ارقام به عنوان حرف آغازین نام متغیر ممنوع است.
- ۵- نام متغیر نباید دارای فضای خالی (Space) باشد.
- ۶- حداکثر طول نام یک متغیر ۳۱ کاراکتر می باشد.
- ۷- در نام گذاری متغیرها، نمیتوان از کلمات رزرو شده زبان مثل if، else، While، int و... به عنوان نام یک متغیر استفاده کرد.
- ۸- استفاده از نام های یکسان برای متغیرهای متعلق به یک محدوده (Scope) ممنوع است.

مثال نام گذاری متغیرها



2xy ×

tot5(×

*Sum ×

H!ght ×

T1+t ×

recteXngel ✓

high!there ×

.pcx ×

int ×

_MyVar ✓

A123 ✓

If ×

Double ×

Var_1 ✓

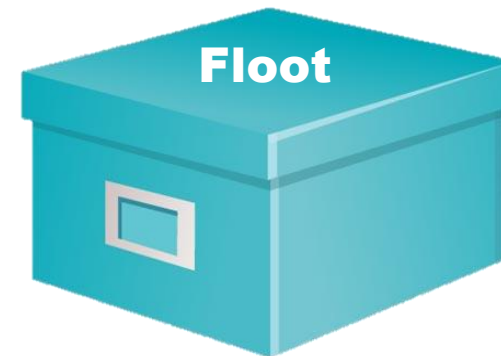
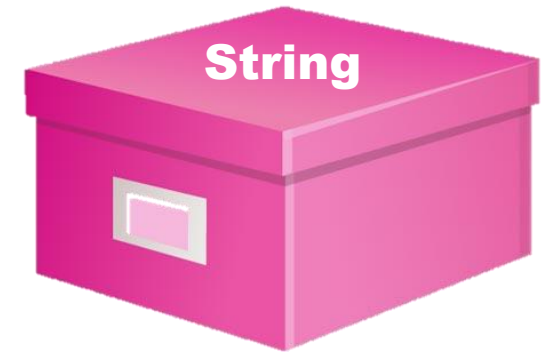
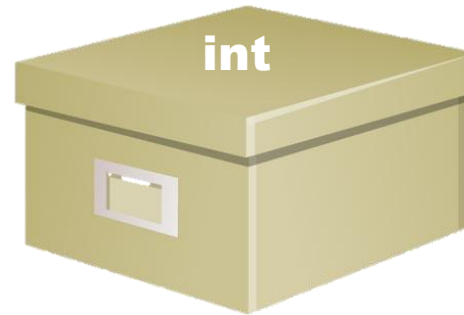
sum.1 ×

Sum ✓

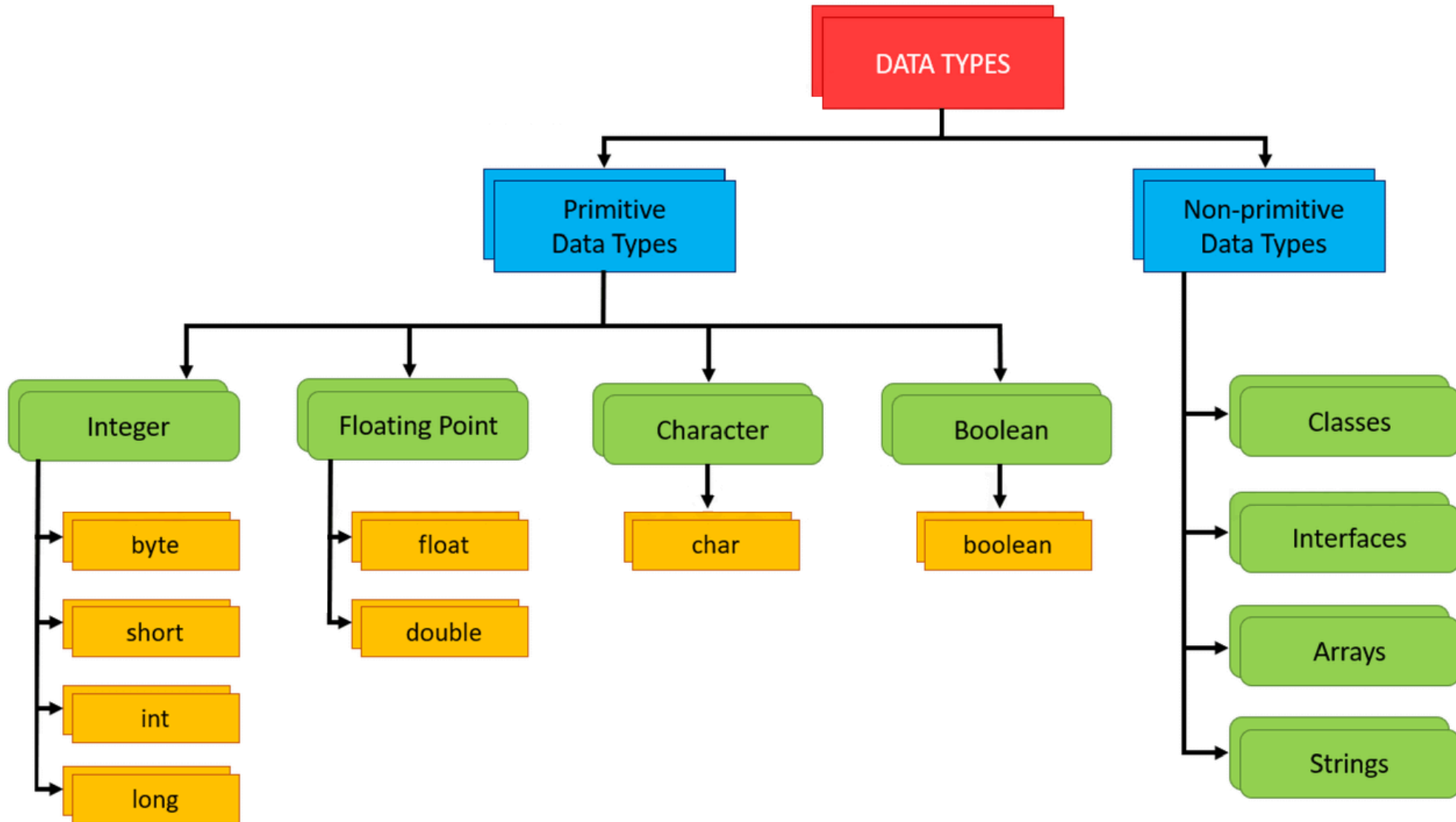
انواع داده در C#



از آنجا که داده‌ها دارای انواع مختلف می‌باشند و متغیرها مکان‌هایی برای ذخیره داده‌ها هستند؛ پس متغیرها نیز دارای نوع می‌باشند. به عبارت دیگر به منظور کار با انواع مختلف داده در برنامه‌ها، نیاز به تعریف انواع مختلف برای متغیرها می‌باشد تا بتوانیم میان آن‌ها تفاوت قائل شویم.



انواع داده در C# – دسته بندی بر اساس اصلی و فرعی بودن



انواع داده اصلی



Type	Description	Range	Suffix
byte	8-bit unsigned integer	0 to 255	
sbyte	8-bit signed integer	-128 to 127	
short	16-bit signed integer	-32,768 to 32,767	
ushort	16-bit unsigned integer	0 to 65,535	
int	32-bit signed integer	-2,147,483,648 to 2,147,483,647	
uint	32-bit unsigned integer	0 to 4,294,967,295	u
long	64-bit signed integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	l

به حرف u در ابتدای برخی از انواع داده ها مثل ushort توجه کنید. این بدان معناست که این نوع فقط شامل اعداد مثبت و صفر هستند.

انواع داده اصلی



Type	Description	Range	Suffix
ulong	64-bit unsigned integer	0 to 18,446,744,073,709,551,615	ul
float	32-bit Single-precision floating point type (7 digits of precision)	-3.402823e38 to 3.402823e38	f
double	64-bit double-precision floating point type (15-16 digits of precision)	-1.79769313486232e308 to 1.79769313486232e308	d
decimal	128-bit decimal type for financial and monetary calculations (28-29 digits of precision)	(+ or -)1.0 x 10e-28 to 7.9 x 10e28	m
char	16-bit single Unicode character	Any valid character, e.g. a, *, \x0058 (hex), or \u0058 (Unicode)	
bool	8-bit logical true/false value	True or False	

انواع داده اصلی




Type	Description	Range	Suffix
bool	8-bit logical true/false value	True or False	
object	Base type of all other types.		
string	A sequence of Unicode characters		
DateTime	Represents date and time	0:00:00am 1/1/01 to 11:59:59pm 12/31/9999	

- نوع char برای ذخیره کاراکترهای یونیکد (Unicode) استفاده می‌شود. کاراکترها باید داخل یک کوتیشن ساده قرار بگیرند مانند: 'a'.
- نوع bool فقط می‌تواند مقادیر درست (true) یا نادرست (false) را در خود ذخیره کند و بیشتر در برنامه‌هایی که دارای ساختار تصمیم‌گیری هستند، مورد استفاده قرار می‌گیرد.
- نوع string برای ذخیره گروهی از کاراکترها مانند یک پیغام استفاده می‌شود. مقادیر ذخیره شده در یک رشته باید داخل دابل کوتیشن قرار گیرند تا توسط کامپایلر به عنوان یک رشته در نظر گرفته شوند، مانند: "Message"

اعلان یا تعریف متغیر

به عملیات تعیین نوع، نام و مقدار اولیه متغیر در زبان های برنامه نویسی اعلان یا تعریف متغیر گفته می شود.

؛ مقدار اولیه متغیر = نام متغیر نوع متغیر

Variable_Type Variable_Name;  int A;

Variable_Type Variable_Name= Initial_value;  int A=5;

تخصیص مقدار به متغیر



• این کار معمولاً با استفاده از عملگر انتساب که در زبان برنامه نویسی C# با علامت = نشان داده می شود، انجام می گردد.

• در اکثر زبان های برنامه نویسی سه روش برای مقداردهی به متغیرها وجود دارد:

۱- مقداردهی در هنگام اعلان متغیر:

```
int X = 7;  
char chr = 'W';  
string str = "Masoud";
```

۲- مقداردهی پس از اعلان متغیر و با استفاده از دستور انتساب:

```
int X;           string str;  
X = 5;          str = "Masoud";
```

۳- با استفاده از دستورات ورودی:

```
int X;  
X = Int32.Parse(Console.ReadLine());
```

```
string str;  
str = Console.ReadLine();
```

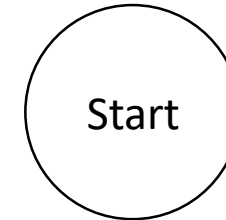
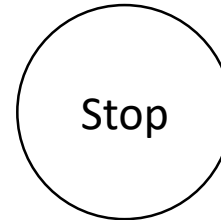

دستورات در پایه سی شارپ



- قاب اصلی برنامه

```
using System;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```



شروع

پایان

خواندن مقادیر از ورودی



```
Console.Read();  
Console.ReadLine();  
Console.ReadKey();
```

A, B ,C ,.....

یک مقدار بخوان

این دستورات مقداری را از ورودی میخوانند. جهت استفاده از مقادیر خوانده شده باید مقدار خوانده شده را به یک متغیر نسبت داد.

```
string str=Console.ReadLine();
```

```
int num =Console.Read();
```

```
ConsoleKeyInfo myKey = Console.ReadKey();
```



```
Console.Write(num + "ABC");
```

```
Console.WriteLine("A\"BC");
```

A, "ABC"

چاپ کن

Escape sequence	Description
<code>\n</code>	Newline. Position the screen cursor to the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the screen cursor to the next tab stop.
<code>\r</code>	Carriage return. Position the screen cursor to the beginning of the current line; do not advance to the next line. Any characters output after the carriage return overwrite the previous characters output on that line.
<code>\\</code>	Backslash. Used to print a backslash character.
<code>\"</code>	Double quote. Used to print a double quote (") character.



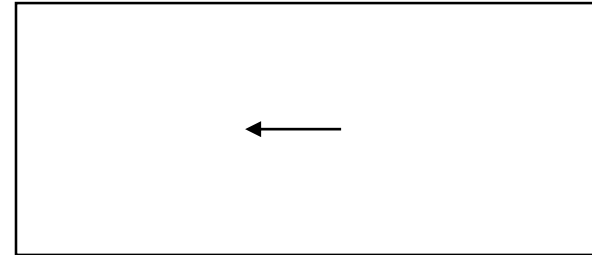
$$A = B + C;$$

$$C = D / F;$$

$$E = A * B;$$

$$G = D - R;$$

$$\text{delta} = B ^ 2 - 4 * A * C;$$



دستورات شرطی

```
if (TestExpression)  
{  
    Body of if statement;  
}
```

statement just below if

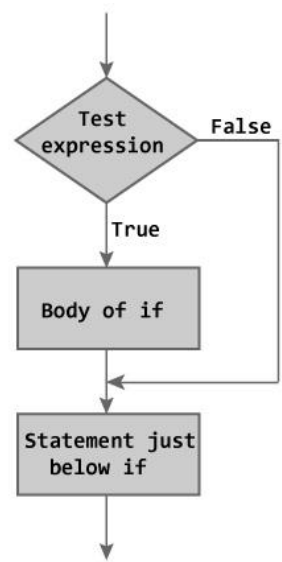


Figure: Flowchart of if Statement

اگر (شرط) آنگاه
دستورات بدنه

Test expression is true

```
int test = 5;  
  
if (test < 10)  
{  
    // codes  
}  
  
// codes after if
```

Test expression is false

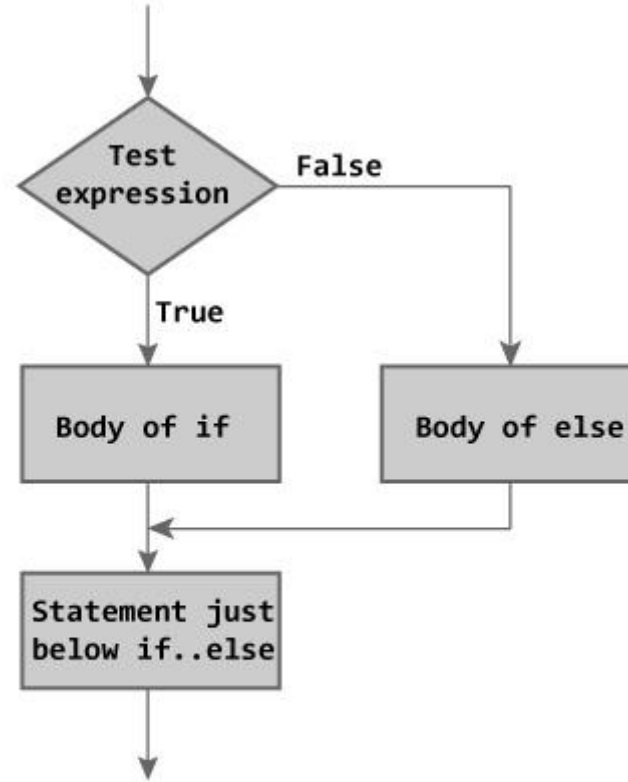
```
int test = 5;  
  
if (test > 10)  
{  
    // codes  
}  
  
// codes after if
```

دستور شرطی دو مرحله ای



```
if (TestExpression)
{
    Body of if statement;
}
else
{
    Body of else
}

statement just below if
```



اگر شرط آنگاه.....

در غیر این صورت

Figure: Flowchart of if...else Statement

دستور شرطی دو مرحله ای



Test expression is true

```
int test = 5;

if (test < 10)
{
    // codes
}
else
{
    // codes
}

// codes after if...else
```

Test expression is false

```
int test = 5;

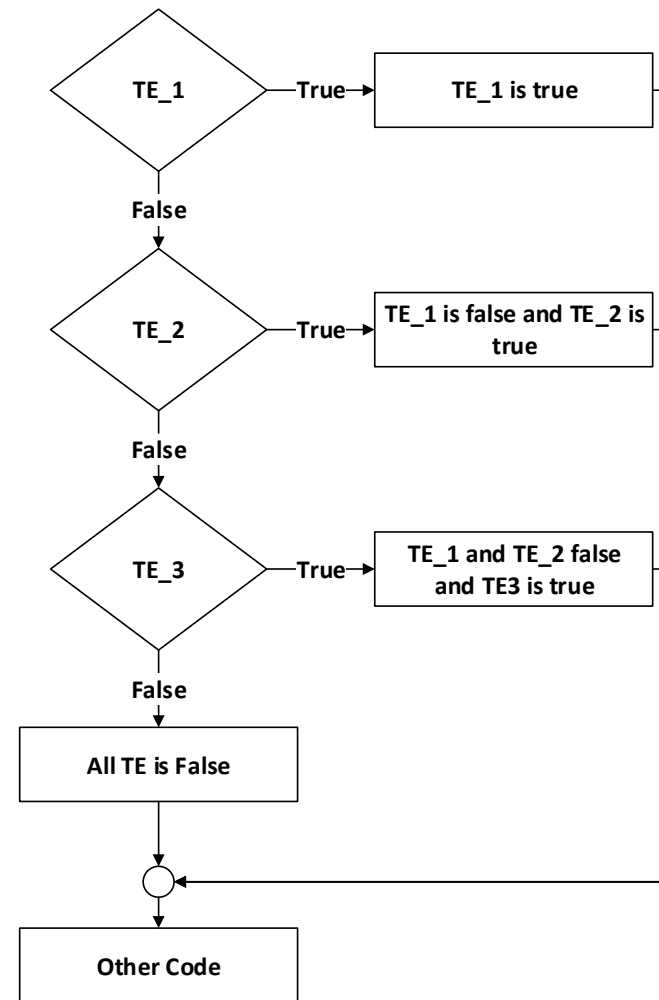
if (test > 10)
{
    // codes
}
else
{
    // codes
}

// codes after if...else
```

دستورات شرطی تو در تو



```
if (testExpression1)
{
    statements to be executed
    if testExpression1 is true
}
else if(testExpression2)
{
    statements to be executed if testExpression1 is false
    and testExpression2 is true
}
else if (testExpression 3)
{
    statements to be executed if testExpression1
    and testExpression2 is false and testExpression3 is true
}
.
.
else
{
    statements to be executed if all
    test expressions are false
}
```



مثال ۱



□ برنامه ای بنویسید که ۳ عدد از ورودی دریافت کرده حاصل جمع دو عدد اول را بر عدد سوم تقسیم نماید.

```
int a, b, c, result;
```

```
Console.Write("Enter Num1:");
```

```
a = Convert.ToInt32(Console.ReadLine());
```

```
Console.Write("Enter Num2:");
```

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

```
Console.Write("Enter Num3:");
```

```
c = Convert.ToInt32(Console.ReadLine());
```

```
result = (a + b);
```

```
result = result / c;
```

```
Console.Write("Result is:");
```

```
Console.WriteLine(result);
```

```
Console.ReadKey();
```

```
result = (a + b)/c;
```

۰- شروع کن.

۱- A و B و C و Result را تعریف کن.

۲- A و B و C را از ورودی دریافت کن.

۳- $Result \leftarrow A+B$

۴- $Result / C$ چاپ کن.

۵- پایان.

مثال ۱ – پاسخ بهینه



□ برنامه ای بنویسید که ۳ عدد از ورودی دریافت کرده حاصل جمع دو عدد اول را بر عدد سوم تقسیم نماید.

```
int a, b, c, result;
Console.Write("Enter Num1:");
a = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter Num2:");
b = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter Num3:");
c = Convert.ToInt32(Console.ReadLine());
result = (a + b)/c;
Console.Write("Result is:" + result.ToString());
Console.ReadKey();
```



```
int a, b, c;
Console.Write("Enter Num1:");
a = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter Num2:");
b = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter Num3:");
c = Convert.ToInt32(Console.ReadLine());
Console.Write("Result is:" + (a + b)/c);
Console.ReadKey();
```

مثال ۲



□ برنامه بنویسید که شعاع دایره ای را از ورودی دریافت کرده محیط و مساحت آن را محاسبه نماید.

$$Area = \pi \times R^2 = 3.14 \times R^2$$

$$Circumference = 2\pi R = 2 \times 3.14 \times R$$

```
Double R , Area, Circ;
Console.Write("Enter a Radius:");
R = Convert.ToInt32(Console.ReadLine());

Area = Math.PI * Math.Pow(R,2);
Circ = 2 * Math.PI * R;

Console.WriteLine("Area is:" + Area);
Console.WriteLine("Circumference is:" + Circ);
Console.ReadKey();
```

۰- شروع کن

۱- R و Area و Circ را تعریف کن.

۲- R را از ورودی دریافت کن.

$$Area \leftarrow 3.14 * R^2 \text{ - ۳}$$

$$Circ = 2 * 3.14 * R \text{ - ۴}$$

۵- Area و Circ را چاپ کن.

۶- پایان

مثال ۳



□ الگوریتمی بنویسید که یک عدد را از ورودی دریافت کرده و در صورت منفی بودن عدد پیام "Negative" و در صورت مثبت بودن عدد پیام "Positive" و در صورت صفر بودن عدد پیام "Zero" را در خروجی چاپ نماید.

```
int N;
```

```
Console.Write("Enter a number:");
```

```
N = Convert.ToInt32(Console.ReadLine());
```

```
if(N==0)
```

```
    Console.WriteLine(N + " is Zero");
```

```
if (N > 0)
```

```
    Console.WriteLine(N + " is Positive");
```

```
if (N < 0)
```

```
    Console.WriteLine(N + " is Negative");
```

```
Console.ReadKey();
```

۰- شروع کن

۱- N را تعریف کن.

۲- N را از ورودی دریافت کن.

۳- اگر $N==0$ بود آنگاه: چاپ کن "Zero"

۴- اگر $N>0$ بود آنگاه: چاپ کن "Positive"

۵- اگر $N<0$ بود آنگاه: چاپ کن "Negative"

۶- پایان

مثال ۳ - پاسخ بهینه



```
int N;  
Console.Write("Enter a number:");  
N = Convert.ToInt32(Console.ReadLine());  
if (N > 0)  
    Console.WriteLine(N + " is Positive");  
else  
    if (N < 0)  
        Console.WriteLine(N + " is Negative");  
    else  
        Console.WriteLine(N + " is Zero");  
Console.ReadKey();
```

۰ - شروع کن

۱ - N را تعریف کن.

۲ - N را از ورودی دریافت کن.

۳ - اگر $N > 0$ بود آنگاه: چاپ کن "Positive"

در غیر این صورت

اگر $N < 0$ بود آنگاه: چاپ کن "Negative"

در غیر این صورت

چاپ کن "Zero"

۴ - پایان

مثال ۴



□ برنامه ای بنویسید که یک عدد را از ورودی دریافت کرده زوج یا فرد بودن آن را مشخص نماید.

□ **قانون ریاضی:** اعداد فرد به اعداد صحیحی گفته می‌شوند که بر ۲ بخش پذیر نباشند. یعنی حاصل تقسیم آنها به دارای باقی مانده یک باشد.

عدد زوج

$$\begin{array}{r|l} 2 & 254 \\ \hline & 127 \\ \hline & 0 \end{array}$$

باقیمانده تقسیم عدد بر ۲ برابر **صفر است** پس عدد زوج است.

عدد فرد

$$\begin{array}{r|l} 2 & 139 \\ \hline & 69 \\ \hline & 1 \end{array}$$

باقیمانده تقسیم عدد بر ۲ برابر **صفر نیست** پس عدد فرد است.

مثال ۴



```
int N;  
Console.Write("Enter a number:");  
N = Convert.ToInt32(Console.ReadLine());  
if (N%2==0)  
    Console.WriteLine(N+" is Even");  
else  
    Console.WriteLine(N+" is Odd");  
Console.ReadKey();
```

۰- شروع کن

۱- N را تعریف کن.

۲- N را از ورودی دریافت کن.

۳- اگر $N \% 2 == 0$ بود آنگاه:

چاپ کن " عدد زوج است " .

در غیر این صورت:

چاپ کن " عدد فرد است " .

۴- پایان.

مثال ۵



□ برنامه ای بنویسید که یکی از اعداد ۰ تا ۶ را از ورودی دریافت کرده نام روز هفته متناظر با آن را در خروجی چاپ نماید.

۰- شروع کن

اگر $N==3$ بود آنگاه چاپ کن "Tuesday"

در غیر این صورت

۱- N را تعریف کن.

اگر $N==4$ بود آنگاه چاپ کن "Wednesday"

در غیر این صورت

۲- N را از ورودی دریافت کن.

۳- اگر $N==0$ بود آنگاه چاپ کن "Saturday"

در غیر این صورت

اگر $N==5$ بود آنگاه چاپ کن "Thursday"

در غیر این صورت

اگر $N==1$ بود آنگاه چاپ کن "Sunday"

در غیر این صورت

اگر $N==6$ بود آنگاه چاپ کن "Friday"

۴- پایان

اگر $N==2$ بود آنگاه چاپ کن "Monday"

پاسخ مثال ۵

```
int N;  
  
Console.Write("Enter a number:");  
  
N = Convert.ToInt32(Console.ReadLine());  
  
if (N == 0)  
    Console.WriteLine("Saturday");  
  
else if (N == 1)  
    Console.WriteLine("Sunday");  
  
else if (N == 2)  
    Console.WriteLine("Monday");  
  
else if (N == 3)  
    Console.WriteLine("Tuesday");
```

```
else if (N == 4)  
    Console.WriteLine("Wednesday");  
  
else if (N == 5)  
    Console.WriteLine("Thursday");  
  
else if (N == 6)  
    Console.WriteLine("Friday");  
  
else  
    Console.WriteLine("Please enter  
a number between 0 and 6.");  
  
Console.ReadKey();
```

مثال ۶



□ برنامه ای بنویسید که سه عدد از ورودی دریافت کرده و تعیین کند آیا می توان با این سه عدد یک مثلث ایجاد کرد یا خیر؟

• راهنمایی: برای ایجاد یک مثلث می بایست رابطه زیر در بین اضلاع آن صادق باشد.

$$\begin{cases} a \leq b + c \\ b \leq a + c \\ c \leq a + b \end{cases}$$

راه حل



$$\text{If } \begin{cases} a \leq b + c \\ \text{و} \\ b \leq a + c \\ \text{و} \\ c \leq a + b \end{cases} \text{ then you can create a triangle}$$

Otherwise the triangle cannot be created



```
int a, b, c;
Console.WriteLine("Triangle making program");
Console.Write("Enter A:");
a = Int32.Parse(Console.ReadLine());
Console.Write("Enter B:");
b = Int32.Parse(Console.ReadLine());
Console.Write("Enter C:");
c = Int32.Parse(Console.ReadLine());

if (a <= b + c)
    if (b <= a + c)
        if (c <= a + b)
            Console.WriteLine("A triangle can be created");
        else
            Console.WriteLine("A triangle can't be created");
    else
        Console.WriteLine("A triangle can't be created");
else
    Console.WriteLine("A triangle can't be created");

Console.ReadKey();
```



AND

پاسخ بهینه مثال ۶

```
int a, b, c;
Console.WriteLine("Triangle making program");
Console.Write("Enter A:");
a = Int32.Parse(Console.ReadLine());
Console.Write("Enter B:");
b = Int32.Parse(Console.ReadLine());
Console.Write("Enter C:");
c = Int32.Parse(Console.ReadLine());
if(a<=b+c && b<=a+c && c<=a+b)
    Console.WriteLine("A triangle can be created");
else
    Console.WriteLine("A triangle can't be created");
Console.ReadKey();
```

مثال ۷



□ برنامه ای بنویسید که سه عدد از ورودی دریافت کرده و تعیین کند آیا می توان با این سه عدد یک مثلث قائم الزاویه ساخت یا خیر؟

• راهنمایی: برای ایجاد یک مثلث قائم الزاویه است به شرط آنکه یکی از روابط زیر بین اضلاع آن صادق باشد.

$$\begin{cases} a^2 = b^2 + c^2 \\ b^2 = a^2 + c^2 \\ c^2 = a^2 + b^2 \end{cases}$$

راه حل



If $\begin{cases} a^2 = b^2 + c^2 \\ \text{یا} \\ b^2 = a^2 + c^2 \\ \text{یا} \\ c^2 = a^2 + b^2 \end{cases}$ then you can create a right triangle



OR

Otherwise the right triangle cannot be created

پاسخ مثال ۷

```
int a, b, c;
Console.WriteLine("Right Triangle detector program");
Console.Write("Enter A:");
a = Int32.Parse(Console.ReadLine());
Console.Write("Enter B:");
b = Int32.Parse(Console.ReadLine());
Console.Write("Enter C:");
c = Int32.Parse(Console.ReadLine());
if (Math.Pow(a, 2) == Math.Pow(b, 2) + Math.Pow(c, 2))
{
    Console.WriteLine("A right triangle can be created");
}
else
{
    if (Math.Pow(b, 2) == Math.Pow(a, 2) + Math.Pow(c, 2))
    {
        Console.WriteLine("A right triangle can be created");
    }
    else
    {
        if (Math.Pow(c, 2) == Math.Pow(a, 2) + Math.Pow(b, 2))
        {
            Console.WriteLine("A right triangle can be created");
        }
        else
        {
            Console.WriteLine("A triangle can't be created");
        }
    }
}
Console.ReadKey();
```



OR



```
int a, b, c;
Console.WriteLine("Right Triangle detector program");
Console.Write("Enter A:");
a = Int32.Parse(Console.ReadLine());
Console.Write("Enter B:");
b = Int32.Parse(Console.ReadLine());
Console.Write("Enter C:");
c = Int32.Parse(Console.ReadLine());

if (
    (Math.Pow(a, 2) == Math.Pow(b, 2) + Math.Pow(c, 2)) ||
    (Math.Pow(b, 2) == Math.Pow(a, 2) + Math.Pow(c, 2)) ||
    (Math.Pow(c, 2) == Math.Pow(a, 2) + Math.Pow(b, 2))
)
    Console.WriteLine("A right triangle can be created");
else
    Console.WriteLine("A triangle can't be created");
Console.ReadKey();
```

مثال ۸



□ برنامه ای بنویسید که مقدار پارامترهای یک معادله درجه دوم را از ورودی دریافت کرده مقدار ریشه های معادله را محاسبه نماید؟

$$D = b^2 - 4 * a * c$$

$$\text{if } D > 0 \implies x_1, x_2 = \frac{-b \pm \sqrt{D}}{2 * a}$$

$$\text{if } D = 0 \implies x_1 = x_2 = \frac{-b}{2 * a}$$

معادله دارای ریشه حقیقی نیست $\implies \text{if } D < 0$

پاسخ مثال ۸

```
Double a, b, c, D, x1, x2;
Console.Write("Enter a:");
a = double.Parse(Console.ReadLine());
Console.Write("Enter b:");
b = double.Parse(Console.ReadLine());
Console.Write("Enter c:");
c = double.Parse(Console.ReadLine());

D = Math.Pow(b, 2) - (4 * a * c);
if(D>0)
{
    x1 = (-b + Math.Sqrt(D)) / (2 * a);
    x2 = (-b - Math.Sqrt(D)) / (2 * a);
    Console.WriteLine("X1="+ x1);
    Console.WriteLine("X2=" + x2);
}
else if(D==0)
{
    x1 = (-b) / (2 * a);
    x2 = x1;
    Console.WriteLine("X1=" + x1);
    Console.WriteLine("X2=" + x2);
}
else
{
    Console.WriteLine("The equation has no root!");
}
Console.ReadKey();
```