

جلسه اول مجازی C#  
برنامه نویسی پیشرفته  
کاردانی نرم ترم یک

مدرس: فاطمه دهقانی فیروزآبادی

## ساختارهای تصمیم

تقریباً همه زبانهای برنامه نویسی به شما اجازه اجرای کد را در شرایط مطمئن می دهند. حال تصور کنید که یک برنامه دارای ساختار تصمیم گیری نباشد و همه کدها را اجرا کند. این حالت شاید فقط برای چاپ یک پیغام در صفحه مناسب باشد ولی فرض کنید که شما بخواهید اگر مقدار یک متغیر با یک عدد برابر باشد سپس یک پیغام چاپ شود آن وقت با مشکل مواجه خواهید شد. سی شارپ راه های مختلفی برای رفع این نوع مشکلات ارائه می دهد. در این بخش با مطالب زیر آشنا خواهید شد :

- دستور if
- دستور if...else
- عملگر سه تایی
- دستور if چندگانه
- دستور if تو در تو
- عملگرهای منطقی
- دستور switch

## دستور if

می‌توان با استفاده از دستور if و یک شرط خاص که باعث ایجاد یک کد می‌شود یک منطق به برنامه خود اضافه کنید. دستور if ساده‌ترین دستور شرطی است که برنامه می‌گوید اگر شرطی برقرار است کد معینی را انجام بده. ساختار دستور if به صورت زیر است :

```
if (condition)  
    code to execute;
```

قبل از اجرای دستور if ابتدا شرط بررسی می‌شود. اگر شرط برقرار باشد یعنی درست باشد سپس کد اجرا می‌شود. شرط یک عبارت مقایسه‌ای است. می‌توان از عملگرهای مقایسه‌ای برای تست درست یا اشتباه بودن شرط استفاده کرد. اجازه بدهید که نگاهی به نحوه استفاده از دستور if در داخل برنامه بیندازیم. برنامه زیر پیغام Hello World را اگر مقدار number کمتر از 10 و Goodbye World را اگر مقدار number از 10 بزرگ‌تر باشد در صفحه نمایش می‌دهد.

```

1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         //Declare a variable and set it a value less than 10
8         int number = 5;
9
10        //If the value of number is less than 10
11        if (number < 10)
12            Console.WriteLine("Hello World.");
13
14        //Change the value of a number to a value which
15        // is greater than 10
16        number = 15;
17
18        //If the value of number is greater than 10
19        if (number > 10)
20            Console.WriteLine("Goodbye World.");
21    }
22 }

```

```

Hello World.
Goodbye World.

```

در خط 8 یک متغیر با نام number تعریف و مقدار 5 به آن اختصاص داده شده است. وقتی به اولین دستور if در خط 11 می‌رسیم برنامه تشخیص می‌دهد که مقدار number از 10 کمتر است یعنی 5 کوچک‌تر از 10 است.

منطقی است که نتیجه مقایسه درست می‌باشد بنابراین دستور if دستور را اجرا می‌کند (خط 12) و پیغام Hello World چاپ می‌شود. حال مقدار number را به 15 تغییر می‌دهیم (خط 16). وقتی به دومین دستور if در خط 19 می‌رسیم برنامه مقدار number را با 10 مقایسه می‌کند و چون مقدار number یعنی 15 از 10 بزرگ‌تر است برنامه پیغام Goodbye World را چاپ می‌کند (خط 20). به این نکته توجه کنید که دستور if را می‌توان در یک خط نوشت :

```
if ( number > 10 ) Console.WriteLine("Goodbye World.");
```

شما می‌توانید چندین دستور را در داخل دستور if بنویسید. کافایت که از یک آکولاد برای نشان دادن ابتدا و انتهای دستورات استفاده کنید. همه دستورات داخل بین آکولاد جز بدنه دستور if هستند. نحوه تعریف چند دستور در داخل بدنه if به صورت زیر است :

```
if (condition)
{
    statement1;
    statement2;
    .
    .
    .
    statementN;
}
```

این هم یک مثال ساده :

```
if (x > 10)
{
    Console.WriteLine("x is greater than 10.");
    Console.WriteLine("This is still part of the if statement.");
}
```

در مثال بالا اگر مقدار x از 10 بزرگ‌تر باشد دو پیغام چاپ می‌شود. حال اگر به عنوان مثال آکولاد را حذف کنیم و مقدار x از 10 بزرگ‌تر نباشد مانند کد زیر :

```
if (x > 10)
    Console.WriteLine("x is greater than 10.");
    Console.WriteLine("This is still part of the if statement. (Really?)");
```

کد بالا در صورتی بهتر خوانده می‌شود که بین دستورات فاصله بگذاریم.

```
if (x > 10)
    Console.WriteLine("x is greater than 10.");

Console.WriteLine("This is still part of the if statement. (Really?)");
```

در مثال بالا اگر مقدار x از 10 بزرگتر باشد دو پیغام چاپ می‌شود. حال اگر به عنوان مثال آکولاد را حذف کنیم و مقدار x از 10 بزرگتر نباشد مانند کد زیر:

```
if (x > 10)
    Console.WriteLine("x is greater than 10.");
    Console.WriteLine("This is still part of the if statement. (Really?)");
```

کد بالا در صورتی بهتر خوانده می‌شود که بین دستورات فاصله بگذاریم.

```
if (x > 10)
    Console.WriteLine("x is greater than 10.");

Console.WriteLine("This is still part of the if statement. (Really?)");
```

می‌بیند که دستور دوم (خط 3) در مثال بالا جز دستور if نیست. اینجاست که چون ما فرض را بر این گذاشته‌ایم که مقدار x از 10 کوچکتر است پس خط (Really?) This is still part of the if statement. چاپ می‌شود. در نتیجه اهمیت وجود آکولاد مشخص می‌شود. به عنوان تمرین همیشه حتی اگر فقط یک دستور در بدنه if داشتید برای آن یک آکولاد بگذارید. فراموش نکنید که از قلم انداختن یک آکولاد باعث به وجود آمدن خطا شده و یافتن آن را سخت می‌کند. یکی از خطاهای معمول کسانی که برنامه نویسی را تازه شروع کرده‌اند قرار دادن سیمیکولن در سمت راست پرانتز if است. به عنوان مثال:

```
if (x > 10);
    Console.WriteLine("x is greater than 10");
```

به یاد داشته باشید که if یک مقایسه را انجام می‌دهد و دستور اجرایی نیست. بنابراین برنامه شما با یک خطای منطقی مواجه می‌شود. همیشه به یاد داشته باشید که قرار گرفتن سیمیکولن در سمت راست پرانتز if به منزله این است که بلوک کد در اینجا به پایان رسیده است. مثالی دیگر در مورد دستور if:

```

using System;

public class Program
{
    public static void Main()
    {
        int firstNumber;
        int secondNumber;

        Console.Write("Enter a number: ");
        firstNumber = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter another number: ");
        secondNumber = Convert.ToInt32(Console.ReadLine());

        if (firstNumber == secondNumber)
        {
            Console.WriteLine("{0} == {1}", firstNumber, secondNumber);
        }
        if (firstNumber != secondNumber)
        {
            Console.WriteLine("{0} != {1}", firstNumber, secondNumber);
        }
        if (firstNumber < secondNumber)
        {
            Console.WriteLine("{0} < {1}", firstNumber, secondNumber);
        }
        if (firstNumber > secondNumber)
        {
            Console.WriteLine("{0} > {1}", firstNumber, secondNumber);
        }
        if (firstNumber <= secondNumber)
        {
            Console.WriteLine("{0} <= {1}", firstNumber, secondNumber);
        }
        if (firstNumber >= secondNumber)
        {
            Console.WriteLine("{0} >= {1}", firstNumber, secondNumber);
        }
    }
}

```

```
Enter a number: 2
Enter another number: 5
2 != 5
2 < 5
2 <= 5
Enter a number: 10
Enter another number: 3
10 != 3
10 > 3
10 >= 3
Enter a number: 5
Enter another number: 5
5 == 5
5 <= 5
5 >= 5
```

ما از عملگرهای مقایسه‌ای در دستور if استفاده کرده‌ایم. ابتدا دو عدد که قرار است با هم مقایسه شوند را به عنوان ورودی از کاربر می‌گیریم. اعداد با هم مقایسه می‌شوند و اگر شرط درست بود پیغامی چاپ می‌شود. به این نکته توجه داشته باشید که شرطها مقادیر بولی هستند، بنابراین شما می‌توانید نتیجه یک عبارت را در داخل یک متغیر بولی ذخیره کنید و سپس از متغیر به عنوان شرط در دستور if استفاده کنید. اگر مقدار year برابر 2000 باشد سپس حاصل عبارت در متغیر isNewMillenium ذخیره می‌شود. می‌توان از متغیر برای تشخیص کد اجرایی بدنه دستور if استفاده کرد خواه مقدار متغیر درست باشد یا نادرست.

```
bool isNewMillenium = year == 2000;

if (isNewMillenium)
{
    Console.WriteLine("Happy New Millenium!");
}
```



## دستور if...else

دستور if فقط برای اجرای یک حالت خاص به کار می‌رود یعنی اگر حالتی برقرار بود کار خاصی انجام شود. اما زمانی که شما بخواهید اگر شرط خاصی برقرار شد یک دستور و اگر برقرار نبود دستور دیگر اجرا شود باید از دستور if else استفاده کنید. ساختار دستور if else در زیر آمده است :

```
if (condition)
{
    code to execute if condition is true;
}
else
{
    code to execute if condition is false;
}
```

از کلمه کلیدی else نمی‌توان به تنهایی استفاده کرد بلکه حتماً باید با if به کار برده شود. اگر فقط یک کد اجرایی در داخل بدنه if و بدنه else دارید استفاده از آکولاد اختیاری است. کد داخل بلوک else فقط در صورتی اجرا می‌شود که شرط داخل دستور if نادرست باشد. در زیر نحوه استفاده از دستور if...else آمده است.

```

1  using System;
2
3  public class Program
4  {
5      public static void Main()
6      {
7          int number = 5;
8
9          //Test the condition
10         if (number < 10)
11         {
12             Console.WriteLine("The number is less than 10.");
13         }
14         else
15         {
16             Console.WriteLine("The number is either greater than or equal to 10.");
17         }
18
19         //Modify value of number
20         number = 15;
21
22         //Repeat the test to yield a different result
23         if (number < 10)
24         {
25             Console.WriteLine("The number is less than 10.");
26         }
27         else
28         {
29             Console.WriteLine("The number is either greater than or equal to 10.");
30         }
31     }
32 }

```

در خط 7 یک متغیر به نام number تعریف کرده‌ایم و در خط 10 تست می‌کنیم که آیا مقدار متغیر number از 10 کمتر است یا نه و چون کمتر است در نتیجه کد داخل بلوک if اجرا می‌شود (خط 12) و اگر مقدار number را تغییر دهیم و به مقداری بزرگتر از 10 تغییر دهیم (خط 20)، شرط نادرست می‌شود (خط 23) و کد داخل بلوک else اجرا می‌شود (خط 29). مانند بلوک if نباید به آخر کلمه کلیدی else سیمیکولن اضافه شود.

عملگر شرطی (?:) در سی شارپ مانند دستور شرطی if...else عمل می‌کند. در زیر نحوه استفاده از این عملگر آمده است:

```
<condition> ? <result if true> : <result if false>
```

عملگر شرطی تنها عملگر سه تایی سی شارپ است که نیاز به سه عملوند دارد، شرط، یک مقدار زمانی که شرط درست باشد و یک مقدار زمانی که شرط نادرست باشد. اجازه بدهید که نحوه استفاده این عملگر را در داخل برنامه مورد بررسی قرار دهیم.

```
public class Program
{
    public static void Main()
    {
        string pet1 = "puppy";
        string pet2 = "kitten";
        string type1;
        string type2;

        type1 = (pet1 == "puppy" ) ? "dog" : "cat";
        type2 = (pet2 == "kitten") ? "cat" : "dog";
    }
}
```

برنامه بالا نحوه استفاده از این عملگر شرطی را نشان می‌دهد. خط یک به صورت زیر ترجمه می‌شود: اگر مقدار pet1 برابر با puppy سپس مقدار dog را در type1 قرار بده در غیر این صورت مقدار cat را type1 قرار بده. خط دو به صورت زیر ترجمه می‌شود: اگر مقدار pet2 برابر با kitten سپس مقدار cat را در type2 قرار بده در غیر این صورت مقدار dog. حال برنامه بالا را با استفاده از دستور if else می‌نویسیم:

```
if (pet1 == "puppy")
    type1 = "dog";
else
    type1 = "cat";
```

هنگامی که چندین دستور در داخل یک بلوک if یا else دارید از عملگر شرطی استفاده نکنید چون خوانایی برنامه را پایین می‌آورد. آموزشکده فنی دختران میبد

## دستور if چندگانه

اگر بخواهید چند شرط را بررسی کنید چکار می‌کنید؟ می‌توانید از چندین دستور if استفاده کنید و بهتر است که این دستورات if را به صورت زیر بنویسید:

```
if (condition)
{
    code to execute;
}
else
{
    if (condition)
    {
        code to execute;
    }
    else
    {
        if (condition)
        {
            code to execute;
        }
        else
        {
            code to execute;
        }
    }
}
```

خواندن کد بالا سخت است. بهتر است دستورات را به صورت تو رفتگی در داخل بلوک else بنویسید. می‌توانید کد بالا را ساده‌تر کنید :

```
if (condition)
{
    code to execute;
}
else if (condition)
{
    code to execute;
}
else if (condition)
{
    code to execute;
}
else
{
    code to execute;
}
```

حال که نحوه استفاده از دستور else if را یاد گرفتید باید بدانید که مانند else if ، else نیز به دستور if وابسته است. دستور else if وقتی اجرا می‌شود که اولین دستور if اشتباه باشد حال اگر else if اشتباه باشد دستور else if بعدی اجرا می‌شود. و اگر آن نیز اجرا نشود در نهایت دستور else اجرا می‌شود.

برنامه زیر نحوه استفاده از دستور else if را نشان می‌دهد :

```

public class Program
{
    public static void Main()
    {
        int choice;

        Console.WriteLine("What's your favorite color?");
        Console.WriteLine("[1] Black");
        Console.WriteLine("[2] White");
        Console.WriteLine("[3] Blue");
        Console.WriteLine("[4] Red");
        Console.WriteLine("[5] Yellow");

        Console.Write("Enter your choice: ");
        choice = Convert.ToInt32(Console.ReadLine());

        if (choice == 1)
        {
            Console.WriteLine("You might like my black t-shirt.");
        }
        else if (choice == 2)
        {
            Console.WriteLine("You might be a clean and tidy person.");
        }
        else if (choice == 3)
        {
            Console.WriteLine("You might be sad today.");
        }
        else if (choice == 4)
        {
            Console.WriteLine("You might be inlove right now.");
        }
        else if (choice == 5)
        {
            Console.WriteLine("Lemon might be your favorite fruit.");
        }
        else
        {
            Console.WriteLine("Sorry, your favorite color is " +
                "not in the choices above.");
        }
    }
}

```

```
What's your favorite color?
```

```
[1] Black  
[2] White  
[3] Blue  
[4] Red  
[5] Yellow
```

```
Enter your choice: 1
```

```
You might like my black t-shirt.
```

```
What's your favorite color?
```

```
[1] Black  
[2] White  
[3] Blue  
[4] Red  
[5] Yellow
```

```
Enter your choice: 999
```

```
Sorry, your favorite color is not in the choices above.
```

خروجی برنامه بالا به متغیر choice وابسته است. بسته به اینکه شما چه چیزی انتخاب می‌کنید پیغامهای مختلفی چاپ می‌شود. اگر عددی که شما تایپ می‌کنید در داخل حالت‌های انتخاب نباشد کد مربوط به بلوک else اجرا می‌شود.

# دستور if تو در تو

اجازه بدهید که نحوه استفاده از دستور if تو در تو را نشان دهیم:

```
1 using System;
2
3 public class Program
4 {
5     public static void Main()
6     {
7         int age;
8         string gender;
9
10        Console.Write("Enter your age: ");
11        age = Convert.ToInt32(Console.ReadLine());
12
13        Console.Write("Enter your gender (male/female): ");
14        gender = Console.ReadLine();
15
16        if (age > 12)
17        {
18            if (age < 20)
19            {
20                if (gender == "male")
21                {
22                    Console.WriteLine("You are a teenage boy.");
23                }
24                else
25                {
26                    Console.WriteLine("You are a teenage girl.");
27                }
28            }
29            else
30            {
31                Console.WriteLine("You are already an adult.");
32            }
33        }
34        else
35        {
36            Console.WriteLine("You are still too young.");
37        }
38    }
39 }
```



```
Enter your age: 18
Enter your gender: male
You are a teenage boy.
Enter your age: 12
Enter your gender: female
You are still too young.
```

اجازه بدهید که برنامه را کالبد شکافی کنیم. ابتدا برنامه از شما درباره ستنان سؤال می‌کند (خط 11). در خط 14 درباره جنستان از شما سؤال می‌کند. سپس به اولین دستور if می‌رسد (خط 16). در این قسمت اگر سن شما بیشتر از 12 سال باشد برنامه وارد بدنه دستور if می‌شود در غیر اینصورت وارد بلوک else (خط 34) مربوط به همین دستور if می‌شود.

حال فرض کنیم که سن شما بیشتر از 12 سال است و شما وارد بدنه اولین if شده‌اید. در بدنه اولین if دو دستور if دیگر را مشاهده می‌کنید. اگر سن کمتر 20 باشد شما وارد بدنه if دوم می‌شوید و اگر نباشد به قسمت else متناظر با آن می‌روید (خط 29). دوباره فرض می‌کنیم که سن شما کمتر از 20 باشد، در اینصورت وارد بدنه if دوم شده و با یک if دیگر مواجه می‌شوید (خط 20). در اینجا جنسیت شما مورد بررسی قرار می‌گیرد که اگر برابر "male" باشد، کدهای داخل بدنه سومین if اجرا می‌شود در غیر اینصورت قسمت else مربوط به این if اجرا می‌شود (خط 24). پیشنهاد می‌شود که از if تو در تو در برنامه کمتر استفاده کنید چون خوانایی برنامه را پایین می‌آورد.

## استفاده از عملگرهای منطقی

عملگرهای منطقی به شما اجازه می‌دهند که چندین شرط را با هم ترکیب کنید. این عملگرها حداقل دو شرط را درگیر می‌کنند و در آخر یک مقدار بولی را برمی‌گردانند. در جدول زیر برخی از عملگرهای منطقی آمده است :

عملگر	تلفظ	مثال	تأثیر
&&	And	$z = (x > 2) \&\& (y < 10)$	مقدار Z در صورتی true است که هر دو شرط دو طرف عملگر مقدارشان true باشد. اگر فقط مقدار یکی از شروط false باشد مقدار z، false خواهد شد.
	Or	$z = (x > 2)    (y < 10)$	مقدار Z در صورتی true است که یکی از دو شرط دو طرف عملگر مقدارشان true باشد. اگر هر دو شرط مقدارشان false باشد مقدار z، false خواهد شد.
!	Not	$z = !(x > 2)$	مقدار Z در صورتی true است که مقدار شرط false باشد و در صورتی false است که مقدار شرط true باشد.

به عنوان مثال جمله  $z = (x > 2) \&\& (y < 10)$  را به این صورت بخوانید: "در صورتی مقدار z برابر true است که مقدار x بزرگ‌تر از 2 و مقدار y کوچک‌تر از 10 باشد در غیر اینصورت false است". این جمله بدین معناست که برای اینکه مقدار کل دستور true باشد باید مقدار همه شروط true باشد. عملگر منطقی OR (||) تأثیر متفاوتی نسبت به عملگر منطقی AND (&&) دارد. نتیجه عملگر منطقی OR برابر true است اگر فقط مقدار یکی از شروط true باشد. و اگر مقدار هیچ یک از شروط true نباشد نتیجه false خواهد شد. می‌توان عملگرهای منطقی AND و OR را با هم ترکیب کرده و در یک عبارت به کار برد مانند :

```
if ( (x == 1) && ( (y > 3) || z < 10) )  
{  
    //do something here  
}
```

در اینجا استفاده از پرانتز مهم است چون از آن در گروه بندی شرطها استفاده می‌کنیم. در اینجا ابتدا عبارت  $(y > 3) \parallel (z < 10)$  مورد بررسی قرار می‌گیرد (به علت تقدم عملگرها). سپس نتیجه آن بوسیله عملگر AND با نتیجه  $(x == 1)$  مقایسه می‌شود. حال بیایید نحوه استفاده از عملگرهای منطقی در برنامه را مورد بررسی قرار دهیم:

```
1  using System;  
2  
3  public class Program  
4  {  
5      public static void Main()  
6      {  
7          int age;  
8          string gender;  
9  
10         Console.WriteLine("Enter your age: ");  
11         age = Convert.ToInt32(Console.ReadLine());  
12  
13         Console.WriteLine("Enter your gender (male/female): ");  
14         gender = Console.ReadLine();  
15  
16         if (age > 12 && age < 20)  
17         {  
18             if (gender == "male")  
19             {  
20                 Console.WriteLine("You are a teenage boy.");  
21             }  
22             else  
23             {  
24                 Console.WriteLine("You are a teenage girl.");  
25             }  
26         }  
27         else  
28         {  
29             Console.WriteLine("You are not a teenager.");  
30         }  
31     }  
32 }
```

```
Enter your age: 18
Enter your gender (male/female): female
You are a teenage girl.
Enter you age: 10
Enter your gender (male/female): male
You are not a teenager.
```

برنامه بالا نحوه استفاده از عملگر منطقی AND را نشان می‌دهد (خط 16). وقتی به دستور if می‌رسید (خط 16) برنامه سن شما را چک می‌کند. اگر سن شما بزرگتر از 12 و کوچکتر از 20 باشد (سنتان بین 12 و 20 باشد) یعنی مقدار هر دو true باشد سپس کدهای داخل بلوک if اجرا می‌شوند. اگر نتیجه یکی از شروط false باشد کدهای داخل بلوک else اجرا می‌شود. عملگر AND عملوند سمت چپ را مورد بررسی قرار می‌دهد. اگر مقدار آن false باشد دیگر عملوند سمت راست را بررسی نمی‌کند و مقدار false را بر می‌گرداند. بر عکس عملگر || عملوند سمت چپ را مورد بررسی قرار می‌دهد و اگر مقدار آن true باشد سپس عملوند سمت راست را نادیده می‌گیرد و مقدار true را بر می‌گرداند.

```
if (x == 2 & y == 3)
{
    //Some code here
}

if (x == 2 | y == 3)
{
    //Some code here
}}
```

نکته مهم اینجاست که شما می‌توانید از عملگرهای & و | به عنوان عملگر بیتی استفاده کنید. تفاوت جزئی این عملگرها وقتی که به عنوان عملگر بیتی به کار می‌روند این است که دو عملوند را بدون در نظر گرفتن مقدار عملوند سمت چپ مورد بررسی قرار می‌دهند. به عنوان مثال حتی اگر مقدار عملوند سمت چپ false باشد عملوند سمت چپ به وسیله عملگر بیتی (&) AND ارزیابی می‌شود. اگر شرطها را در برنامه ترکیب کنید استفاده از عملگرهای منطقی AND (&&) و OR (||) به جای عملگرهای بیتی (&) AND و (|) OR بهتر خواهد بود. یکی دیگر از عملگرهای منطقی عملگر (!) NOT است که نتیجه یک عبارت را

خنثی یا منفی می‌کند. به مثال زیر توجه کنید:

```
if (!(x == 2))
{
    Console.WriteLine("x is not equal to 2.");
}
```

اگر نتیجه عبارت `x == 2` برابر `false` باشد عملگر `!` آن را `True` می‌کند.

## دستور Switch

در سی شارپ ساختاری به نام `switch` وجود دارد که به شما اجازه می‌دهد که با توجه به مقدار ثابت یک متغیر چندین انتخاب داشته باشید. دستور `switch` معادل دستور `if` تو در تو است با این تفاوت که در دستور `switch` متغیر فقط مقادیر ثابتی از اعداد، رشته‌ها و یا کاراکترها را قبول می‌کند. مقادیر ثابت مقادیری هستند که قابل تغییر نیستند. در زیر نحوه استفاده از دستور `switch` آمده است :

ابتدا یک مقدار در متغیر `switch` که در مثال بالا `testVar` است قرار می‌دهید. این مقدار با هر یک از عبارتهای `case` داخل بلوک `switch` مقایسه می‌شود. اگر مقدار متغیر با هر یک از مقادیر موجود در دستورات `case` برابر بود کد مربوط به آن `case` اجرا خواهد شد. به این نکته توجه کنید که حتی اگر تعداد خط کدهای داخل دستور `case` از یکی بیشتر باشد نباید از آکولاد استفاده کنیم. آخر هر دستور `case` با کلمه کلیدی `break` تشخیص داده می‌شود که باعث می‌شود برنامه از دستور `switch` خارج شده و دستورات بعد از آن اجرا شوند. اگر این کلمه کلیدی از قلم بیوفتد برنامه با خطا مواجه می‌شود. دستور `switch` یک بخش `default` دارد. این دستور در صورتی اجرا می‌شود که مقدار متغیر با هیچ یک از مقادیر دستورات `case` برابر نباشد. دستور `default` اختیاری است و اگر از بدنه `switch` حذف شود هیچ اتفاقی نمی‌افتد. مکان این دستور هم مهم نیست اما بر طبق تعریف آن را در پایان دستورات می‌نویسند. به مثالی در مورد دستور `switch` توجه کنید :

```

public class Program
{
    public static void Main()
    {
        int choice;

        Console.WriteLine("What's your favorite pet?");
        Console.WriteLine("[1] Dog");
        Console.WriteLine("[2] Cat");
        Console.WriteLine("[3] Rabbit");
        Console.WriteLine("[4] Turtle");
        Console.WriteLine("[5] Fish");
        Console.WriteLine("[6] Not in the choices");
        Console.Write("\nEnter your choice: ");

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

        switch (choice)
        {
            case 1:
                Console.WriteLine("Your favorite pet is Dog.");
                break;
            case 2:
                Console.WriteLine("Your favorite pet is Cat.");
                break;
            case 3:
                Console.WriteLine("Your favorite pet is Rabbit.");
                break;
            case 4:
                Console.WriteLine("Your favorite pet is Turtle.");
                break;
            case 5:
                Console.WriteLine("Your favorite pet is Fish.");
                break;
            case 6:
                Console.WriteLine("Your favorite pet is not in the choices.");
                break;
            default:
                Console.WriteLine("You don't have a favorite pet.");
                break;
        }
    }
}

```

```
Enter your choice: 2
Your favorite pet is Cat.
What's your favorite pet?
[1] Dog
[2] Cat
[3] Rabbit
[4] Turtle
[5] Fish
[6] Not in the choices

Enter your choice: 99
You don't have a favorite pet.
```

برنامه بالا به شما اجازه انتخاب حیوان مورد علاقه‌تان را می‌دهد. به اسم هر حیوان یک عدد نسبت داده شده است. شما عدد را وارد می‌کنید و این عدد در دستور switch با مقادیر case مقایسه می‌شود و با هر کدام از آن مقادیر که برابر بود پیغام مناسب نمایش داده خواهد شد. اگر هم با هیچ کدام از مقادیر case ها برابر نبود دستور default اجرا می‌شود. یکی دیگر از ویژگیهای دستور switch این است که شما می‌توانید از دو یا چند case برای نشان داده یک مجموعه کد استفاده کنید. در مثال زیر اگر مقدار number عدد 1، 2 یا 3 باشد یک کد اجرا می‌شود. توجه کنید که case ها باید پشت سر هم نوشته شوند.

```
switch(number)
{
    case 1:
    case 2:
    case 3:
        Console.WriteLine("This code is shared by three values.");
        break;
}
```

همانطور که قبلاً ذکر شد دستور switch معادل دستور if تو در تو است. برنامه بالا را به صورت زیر نیز می‌توان نوشت :

```
if (choice == 1)
    Console.WriteLine("Your favorite pet is Dog.");
else if (choice == 2)
    Console.WriteLine("Your favorite pet is Cat.");
else if (choice == 3)
    Console.WriteLine("Your favorite pet is Rabbit.");
else if (choice == 4)
    Console.WriteLine("Your favorite pet is Turtle.");
else if (choice == 5)
    Console.WriteLine("Your favorite pet is Fish.");
else if (choice == 6)
    Console.WriteLine("Your favorite pet is not in the choices.");
else
    Console.WriteLine("You don't have a favorite pet.");
```



کد بالا دقیقاً نتیجه‌ای مانند دستور switch دارد. دستور default معادل دستور else می‌باشد. حال از بین این دو دستور (switch و if else) کدامیک را انتخاب کنیم. از دستور switch موقعی استفاده می‌کنیم که مقداری که می‌خواهیم با دیگر مقادیر مقایسه شود ثابت باشد. مثلاً در مثال زیر هیچگاه از switch استفاده نکنید.

```
int myNumber = 5;
int x = 5;

switch (myNumber)
{
    case x:
        Console.WriteLine("Error, you can't use variables as a value" +
            " to be compared in a case statement.");
        break;
}
```

مشاهده می‌کنید که با اینکه مقدار x عدد 5 است و به طور واضح با متغیر myNumber مقایسه شده است برنامه خطا می‌دهد چون x یک ثابت نیست بلکه یک متغیر است یا به زبان ساده‌تر، قابلیت تغییر را دارد. اگر بخواهید از x استفاده کنید و برنامه خطا ندهد باید از کلمه کلیدی const به صورت زیر استفاده کنید.

```
int myNumber = 5;
const int x = 5;

switch (myNumber)
{
    case x:
        Console.WriteLine("Error has been fixed!");
        break;
}
```

از کلمه کلیدی `const` برای ایجاد ثابتها استفاده می‌شود. توجه کنید که بعد از تعریف یک ثابت نمی‌توان مقدار آن را در طول برنامه تغییر داد. به یاد داشته باشید که باید ثابتها را حتماً مقداردهی کنید. دستور `switch` یک مقدار را با مقادیر `case` ها مقایسه می‌کند و شما لازم نیست که به شکل زیر مقادیر را با هم مقایسه کنید :

```
switch (myNumber)
{
    case x > myNumber:
        Console.WriteLine("switch staments can't test if a value is less than " +
            "or greater than the other value.");
        break;
}
```

# تمرین

(لازم به ذکر است که حل تمرین ها فعالیت کلاسی محسوب می شود و تا زمانیکه حل آن توسط مدرس  
ارایه نشده فرصت تحویل دارید که می توانید به شخصی بنده ارسال کنید)

\*جدول زیر را به کمک دستور if پیاده سازی کنید.

محدوده X	خروجی
۵۰ یا کمتر	A
۷۰ - ۵۱	B
۹۰ - ۷۱	C
۹۱ به بالا	D

\* دستور if زیر را به کمک عملگر شرطی بنویسید.

```
if(a>0 && a<10)
{
    a=a+1;
    c=c+a;
}
else
    c=c+a/b;
```

- public static void main(String[] args) {
- int x = 10;
- if (x == 10) {
- Console.WriteLine("Value of X is 10");
- } else if (x == 20) {
- Console.WriteLine("Value of X is 20");
- } else if (x == 30) {
- Console.WriteLine("Value of X is 30");
- } else {
- Console.WriteLine("This is else statement");
- }
- }

\* خروجی برنامه فوق چیست؟ برنامه فوق را trace کنید؟