

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

حساب (Arithmetic)

مدرس : مهندس افشین رفوآ

حساب (Arithmetic)

مقادیر مطلق

سیستم عدد اعشاری (**decimal**) از منفی بی نهایت آغاز شده و به مثبت بی نهایت ختم می شود. به عبارت دیگر، بسته به موقعیت آن ها (این که بعد از صفر قرار می گیرند یا قبل آن) اعداد یا منفی هستند یا مثبت. البته به جز عدد 0 که خنثی تلقی می گردد. در برخی عملیات عدد حتماً باید مثبت باشد، حتی اگر در قالب منفی ارائه شده باشد. مقدار مطلق عدد **x** همان **x** است، البته در صورتی که عدد مورد نظر مثبت باشد. چنانچه عددی منفی بود، مقدار مطلق آن معادل مثبت آن عدد محسوب می شود. برای مثال، مقدار مطلق 12 همان 12 هست، این در حالی است که مقدار مطلق -12، 12 است.

برای به دست آوردن مقدار مطلق یک عدد، کلاس **Math** مجهز به متدی است به نام **Abs**، که در نسخه های متعدد اضافه بار گذاری (**overload**) شده است. دستور زبان (**syntax**) آن ها به شرح زیر است.

```
public static sbyte Abs(sbyte value);
public static short Abs(short value);
public static int Abs(int value);
public static float Abs(float value);
public static double Abs(double value);
public static long Abs(long value);
public static decimal Abs(decimal value);
```

متد فوق آرگومانی می گیرد که باید مقدار مطلق آن را به دست آورد :

```
using System;
class Program
{
    static int Main()
    {
        int number = -6844;

        Console.WriteLine("Original Value = {0}",
number);
        Console.WriteLine("Absolute Value = {0}\n",
Math.Abs(number));
        return 0;
    }
}
```

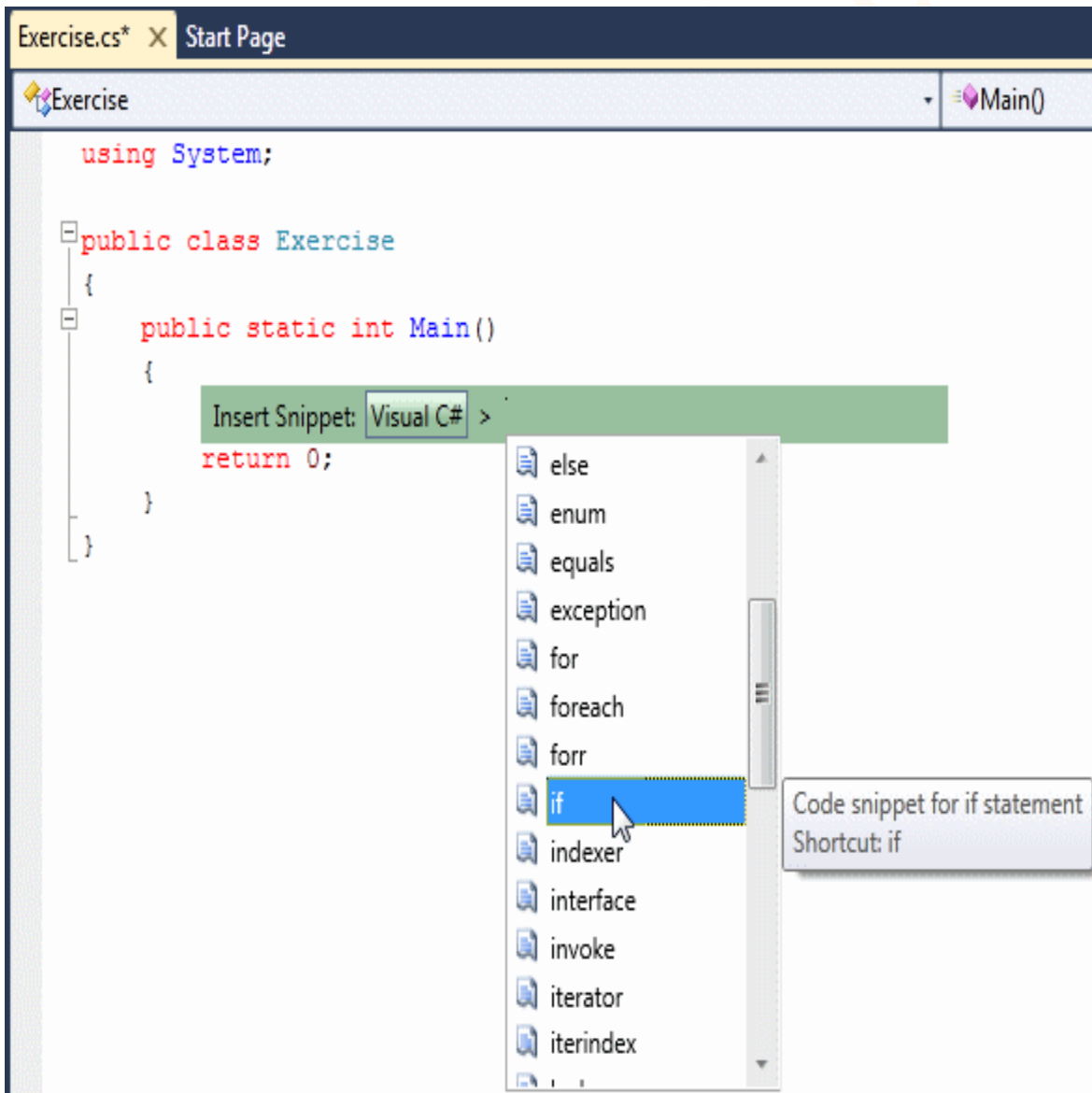
```
}  
}
```

نتیجه ی زیر را بدست می دهد.

```
Original Value = -6844  
Absolute Value = 6844  
Press any key to continue...
```

سقف (بیشترین مقدار) یک عدد

عددی با ممیز اعشار مثل 12.155 را در نظر بگیرید. این عدد بین 12 integer و 13 integer قرار دارد.



به همین نحو، عددی مثل **-24.06** را در نظر بگیرید که منفی بوده و بین اعداد **-24** و **-25** قرار می گیرد. لازم به ذکر نیست که **-24** بزرگ تر است.

در علم حساب سقف یک عدد، نزدیک ترین عدد صحیح (**integer**) هست که از عدد مورد نظر بزرگ تر باشد. در مثال اول سقف **12.155**، عدد **13** است زیرا **13** نزدیک ترین عدد صحیح بزرگتر از (یا برابر با) **12.155** می باشد. حال سقف عدد **-24.06**، **-24** می باشد.

برای به دست آوردن سقف یک عدد، کلاس **Math** مجهز به متدی است به نام **Ceiling** که در دو نسخه اضافه بارگذاری شده است. دستور نحوی این دو نسخه به صورت زیر می باشد.

```
public static double Ceiling(double a);  
public static decimal Ceiling(decimal d);
```

متد بالا به عنوان آرگومان یک متغیر عدد با ممیز اعشار می گیرد که سقف آن را باید به دست آورد.

مثال

```
using System;  
class Program  
{  
    static int Main()  
    {  
        double value1 = 155.55; double value2 = -24.06;  
        Console.WriteLine("The ceiling of {0} is {1}",  
            value1, Math.Ceiling(value1));  
        Console.WriteLine("The ceiling of {0} is {1}\n",  
            value2, Math.Ceiling(value2));  
        return 0;  
    }  
}
```

نتیجه

```
The ceiling of 155.55 is 156  
The ceiling of -24.06 is -24  
Press any key to continue...
```

جدا از کلاس **Math**، ساختار **Double** پیاده سازی خودش از متد نام برده را با دستور نحوی زیر ارائه می دهد.

```
public static decimal Ceiling(decimal d);
```

کف (کمترین مقدار) یک عدد

دو عدد اعشاری 128.44 و -36.72 را در نظر بگیرید. 128.44 بین دو عدد 128 و 129 قرار دارد که 128 عدد کوچکتر محسوب می شود. -36.72 بین دو عدد -37 و -36 قرار گرفته که -37 عدد کوچکتر محسوب می شود. کمترین یا پایین ترین ولی در عین حال نزدیک ترین مقدار **integer** یک عدد را کف آن عدد می گویند.

برای بدست آوردن کمترین مقدار یک عدد کلاس **Math**، متد **Floor()** را ارائه می دهد. این متد در دو نسخه ی زیر اضافه بارگذاری می شود.

```
public static double Floor(double d);  
public static decimal Floor(decimal d);
```

متد **Floor()** مقدار مربوط را به عنوان آرگومان گرفته و عدد صحیحی (**integer**) را برمی گرداند که کوچکتر از (یا برابر با) آرگومان نام برده باشد.

مثال

```
using System;  
  
class Program  
{  
    static int Main()  
    {  
        double value1 = 1540.25;  
        double value2 = -360.04;  
  
        Console.WriteLine("The floor of {0} is {1}",  
            value1, Math.Floor(value1));  
        Console.WriteLine("The floor of {0} is {1}\n",  
            value2, Math.Floor(value2));  
        return 0;  
    }  
}
```

نتیجه ی زیر حاصل می گردد.

```
The floor of 1540.25 is 1540
The floor of -360.04 is -361
Press any key to continue...
```

به جای استفاده از کلاس **Math**، می توان ساختار **Double** را به کار برد که متدی ویژه ی بدست آوردن کم ترین مقدار (کف) عدد اعشاری دارد. دستور نحوی آن به شکل زیر است :

```
public static decimal Ceiling(decimal d);
```

توان یک عدد

در **#C** توان با فرمول زیر انجام می شود :

```
ReturnValue = xy
```

کلاس **Math** برای انجام عملیات توان، متد **Pow** را در اختیار شما قرار می دهد که دستور نحوی آن به شکل زیر است :

```
public static double Pow(double x, double y);
```

این متد دو آرگومان می گیرد. آرگومان اول، **x**، عدد پایه است که باید ارزیابی شود. آرگومان دوم، **y**، که توان نامیده می شود، آرگومان **x** را به توان آرگومان **y** می برد.

مثال

```
using System;
class Program
{
    static int Main()
    {
        const double source = 25.38;
        const double exp = 3.12;
        double result = Math.Pow(source, exp);
        Console.WriteLine("Pow({0}, {1}) = {2}\n",
            source, exp, result);
        return 0;
    }
}
```

نتیجه

```
Pow(25.38, 3.12) = 24099.8226934415
Press any key to continue...
```

مقدار نمایی (the Exponential)

می توان مقدار نمایی یک عدد را محاسبه کرد. کلاس **Math**، متد **Exp()** را برای این منظور در نظر گرفته است. دستور نحوی آن به شکل زیر می باشد.

```
public static double Exp (double d);
```

مثال

```
using System;

class Program
{
    static int Main()
    {
        Console.WriteLine("The exponential of {0} is
{1}",
709.78222656,
Math.Exp(709.78222656));
        return 0;
    }
}
```

نتیجه ی زیر به دست می آید :

```
The exponential of 709.78222656 is 1.79681906923757E+308
Press any key to continue...
```

چنانچه مقدار **x** از عدد (تقریبی) **-708.395996093** کوچکتر بود، مقدار صفر برگردانده می شود و وضعیت پاریز (**underflow**) رخ می دهد. در صورتی که مقدار آرگومان **x** بزرگتر از عدد (تقریبی) **709.78222656** بود، (نتیجه) وضعیت سرریز (**overflow**) رخ می دهد.

لگاریتم طبیعی

برای محاسبه لگاریتم طبیعی یک عدد، می توان متد **Math.Log()** را فراخواند. متد مزبور به دو صورت زیر ارائه می شود. دستور نحوی یکی از آن دو

```
public static double Log(double d);
```

مثال

```
using System;
class Program
{
    static int Main()
    {
        double log = 12.48D;
        Console.WriteLine("Log of {0} is {1}", log,
Math.Log(log));
        return 0;
    }
}
```

نتیجه ی زیر حاصل می شود.

```
Log of 12.48 is 2.52412736294128
Press any key to continue...
```

لگاریتم پایه ی 10

متد **Math.Log10()** لگاریتم پایه ی 10 یک عدد را محاسبه می کند. دستور نحوی متد نام برده به صورت زیر است.

```
public static double Log10(double d);
```

عددی که ارزیابی می شود به عنوان آرگومان ارسال می گردد. متد، لگاریتم مورد نظر را با فرمول زیر بر مبنای 10 بر می گرداند.

$$y = \log_{10}x$$

که معادل

$$x = 10^y$$

مثال

```
using System;
class Program
{
    static int Main()
    {
```



```

        double log10 = 12.48D;
        Console.WriteLine("Log of {0} is {1}", log10,
Math.Log10(log10));

        return 0;
    }
}

```

نتیجه ی زیر حاصل می گردد.

```

Log of 12.48 is 1.09621458534641
Press any key to continue...

```

لگاریتم به هر مبنایی

متد **Math.Log** () نسخه ی دیگری فراهم می کند که دستور نحوی آن بدین صورت است.

```

public static double Log(double a, double newBase);

```

متغیری که مقدار لگاریتمی آن محاسبه می شود، به عنوان اولین آرگومان به متد ارسال می گردد. آرگومان دوم به شما اجازه می دهد مبنای (لگاریتم) دلخواه خود را تعیین کنید. چنین متدی از فرمول زیر استفاده می کند.

$$Y = \log_{\text{NewBase}} x$$

که با فرمول زیر یکی است.

$$x = \text{NewBase}^y$$

مثال

```

using System;
class Program
{
    static int Main()
    {
        double logN = 12.48D;
        Console.WriteLine("Log of {0} is {1}", logN,
Math.Log(logN, 4));
        return 0;
    }
}

```

نتیجه

```
Log of 12.48 is 1.82077301454376
Press any key to continue...
```

ریشه ی دوم / جذر

می توان ریشه ی مربع یک عدد مثبت اعشاری را به دست آورد. برای این منظور، کلاس **Math** مجهز به متدی است به نام **Sqrt** که دستور نحوی آن به شکل زیر می باشد.

```
public static double Sqrt(double d);
```

متد بالا یک آرگومان به عنوان عدد با ممیز شناور مثبت می گیرد. پس از محاسبه، متد ریشه ی مربع (جذر) **x** را برمی گرداند.

```
using System;
class Program
{
    static int Main()
    {
        double sqrt = 8025.73D;
        Console.WriteLine("The square root of {0} is
{1}", sqrt, Math.Sqrt(sqrt));
        return 0;
    }
}
```

نتیجه

```
The square root of 8025.73 is 89.5864387058666
Press any key to continue...
```