

# به نام خالق هستی بخش

## راهنمای استفاده از هدر فایل Console.h

با عرض سلام و خسته نباشید خدمت تمام عزیزان ، امروز می خواهیم هدر فایلی را در C++ به شما معرفی کنیم که کار شما را در مواجهه با جریان ورودی و خروجی و استفاده از محیط کنسول ساده تر می کند در واقع ما با ساخت این هدر فایل که بیشتر جنبه آموزشی دارد سعی کرده ایم کلاس Console زبان C# را در زبان C++ پیاده سازی کنیم تا همانطور که در آنجا می توانید با متد های کلاس Console به سادگی با برنامه های تحت Console کار کنید در C++ هم اینکار تا حدودی ساده تر شود مثلاً همانطور که می دانید در حالت عادی برای چاپ "Hello World" در صفحه کنسول باید از دستور زیر استفاده کنید:

```
cout << "Hello Wolrd" ;
```

حال ما می خواهیم روشی را به شما معرفی کنیم تا از پس قادر باشید مانند برنامه های C# با ورودی و خروجی کار کنید یعنی به عنوان مثال همین عبارت بالا را به صورت زیر بازنویسی کنید:

```
Console.Write("Hello World");
```

خوب برای شروع کار ابتدا فایل Console.h را در کنار فایل برنامه خود قرار بدهید و سپس آن را در برنامه خود include نمایید دقت داشته باشید که با این کار هدر فایل iostream و نیز دستور using namespace std; به طور

خودکار اضافه می شود. و نیازی نیست که شما این دستورات را داخل برنامه خود وارد نمایید.

حال به شیوه کار با این کتابخانه می پردازیم:

## قسمت اول

### دستورات مربوط به Console

#### 1- Console.Write()

این دستور معادل دستور cout در حالت معمول است، و موجب چاپ عبارتی در خروجی می شود.

#### 2- Console.WriteLine()

این دستور نیز معادل استفاده از دستور cout همراه با رد کننده خط می باشد یعنی Console.WriteLine(str) معادل cout << str << endl می باشد.

#### 3- Console.ReadLine()

این تابع معادل دستور cin در حالت معمولی می باشد و به دو شکل می توانید از آن استفاده کنید:

1- بدون هیچگونه آرگومان و در حالت ساده Console.ReadLine() که دقیقاً معادل cin می باشد.

دستورات زیر معادل اند:

```
string str;  
cin >> str ;
```

یا:

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

2- با استفاده از یک آرگومان آن:

Console.ReadLine(int size)

در این حالت شما می توانید مشخص کنید تعداد مشخصی از کاراکتر های ورودی داخل رشته قرار گیرد که باید مقدار مورد نظر خود را بجای متغیر size قرار دهید، مثلا در پایین ما تعیین می کنیم صرفا 10 کاراکتر ابتدایی را در رشته ذخیره کنیم:

```
string str;  
cin.get(str,10);  
cout << str ;
```

معادل:

```
string str;  
str = Console.ReadLine(10);  
Console.Write(str);
```

3- با استفاده از تمام آرگومان های آن:

`Console.ReadLine(int size, char sep)`

در این حالت شما قادر خواهید بود تعداد خاصی از کاراکتر ها را از ورودی را بخواهید که آن را با size مشخص می کنید همچنین می توانید یک کاراکتر هم در نظر بگیرید که از آنجا به بعد دیگر دریافت رشته ادامه پیدا نکند که آن را در قسمت sep مشخص می کنید. بهاین معنا که حداکثر به میزان size کاراکتر دریافت کنیم و اگر قبل از پر شدن ظرفیت به کاراکتر sep رسیده باشیم از آنجا به بعد حساب نشود. دو مورد پایین باهم معادل اند:

```
string str;
```

```
cin.get(str, 10, 'a');
```

```
cout << str;
```

معادل:

```
string str;
```

```
str = Console.ReadLine(10,'a');
```

```
Console.Write(str);
```

**نکته مهم: در هر حالتی که از این تابع استفاده کنید مقدار برگشتی آن یک مقدار رشته ای می باشد.**

#### 4- Console.Clear():

با استفاده از این تابع قادر خواهید بود صفحه خروجی (کنسول) را پاک کنید.

مثال:

```
Console.Clear();
```

#### 5- Console.SetColors():

این تابع برای تنظیم رنگ متن و زمینه به کار می رود:

```
Console.SetColors(char Back, char Fore);
```

با دادن مقدار به Back رنگ زمینه و Fore رنگ متن را تنظیم کنید.

مقادیر را به صورت Color.ColorName وارد کنید که ColorName یکی از موارد زیر می باشد:

Black	سیاه
Blue	آبی
Green	سبز
Red	قرمز
Purple	بنفش
Yellow	زرد
White	سفید
Gray	خاکستری
LightBlue	آبی روشن
LightGreen	سبز روشن
LightRed	قرمز روشن
LightPurple	بنفش روشن
LightYellow	زرد روشن
BrightWhite	سفید درخشان

برای یادگیری بهتر مطلب به مثال زیر توجه کنید:

می خواهیم رنگ زمینه را زرد و رنگ متن را آبی کنیم:

```
Console.SetColors(Color.Yellow, Color.Blue);
```

حال که با کلاس Console آشنا شدید و مشاهده کردید چگونه با این کلاس قادرید عملیات مربوط به ورودی و خروجی را از نظر مفهومی ساده تر کنید می خواهیم به شما کلاس دیگری را در این هدر فایل معرفی کنیم:

## قسمت دوم

دستورات مربوط به Random

با استفاده از این کلاس به سادگی قادر خواهید بود اعداد تصادفی تولید نمایید. تنها تابع (یا بهتر بگوییم متد) در این کلاس تابع Next() می باشد. که شما به سه شکل می توانید از آن استفاده کنید:

1- بدون هیچگونه آرگومان ورودی:

اگر بدون هیچگونه آرگومان ورودی از این تابع استفاده شود در واقع این تابع به طور تصادفی یکی از اعداد 0 یا 1 را باز خواهد گردانند. مقدار بازگشتی این تابع از نوع صحیح و یا int می باشد.

مثال:

```
int randNum = Random.Next();  
Console.Write(randNum);
```

2- با استفاده از یک آرگومان ورودی:

شکل تابع: Random.Next(int num)

با فراخوانی این تابع و ورود آرگومان num مقدار بازگشتی عدد صحیحی بین صفر تا مقدار num می باشد که می تواند شامل صفر هم باشد.

مثال :

برنامه پایین با هر بار اجرا عددی بین صفر و 10 را بر روی صفحه چاپ می کند:

```
int randNum = Random.Next(10);  
Console.Write(randNum);
```

3- با استفاده از هر دو آرگومان ورودی:

شکل تابع: Random.Next(int from, int to)



با فراخوانی این تابع و ورود آرگومان from و to مقدار بازگشتی عدد صحیحی بین from تا مقدار to می باشد که می تواند شامل خود from یا to هم باشد.

مثال :

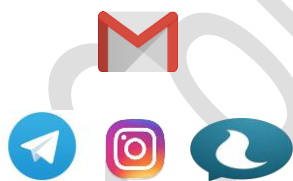
برنامه پایین با هر بار اجرا عددی بین 2 و 10 را بر روی صفحه چاپ می کند:

```
int randNum = Random.Next(2,10);
```

```
Console.Write(randNum);
```

ممنون از اینکه برای مطالعه این راهنما وقت گذاشتید. لطفا هر گونه پیشنهاد برای بهبود هدر فایل گفته شده را حتما برای ما ارسال کنید.

ارتباط با ما:



mahdymorady521@gmail.com

@mahdi\_moradi19

پایان

Console.h

دانشگاه