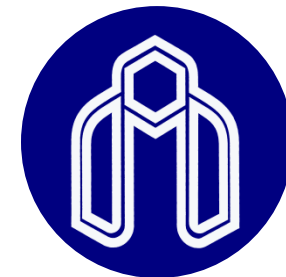


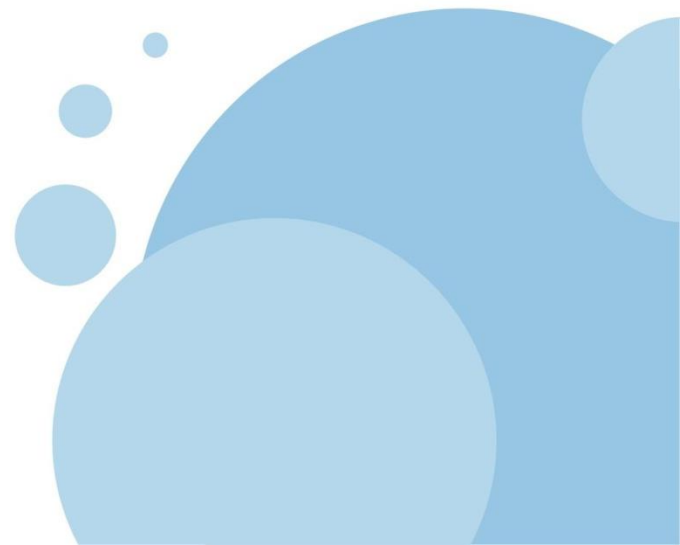
دانشگاه صنعتی شاهرود، مرکز آموزش‌های الکترونیکی



دانشگاه صنعتی شاهرود  
مرکز آموزش‌های الکترونیکی

# مبانی کامپیوتر و برنامه نویسی

جلسه سوم: مقدمات برنامه نویسی با ++C





دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## { مبانی کامپیوتر و برنامه‌نویسی (فهرست مطالب) }

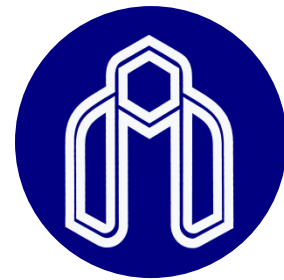
- ۱- مقدمه
- ۲- آماده‌سازی مقدمات
- ۳- شروع کار با C++
- ۴- عملگر خروجی
- ۵- لیترال‌ها و کاراکترها
- ۶- متغیرها و تعریف آن‌ها
- ۷- مقداردهی اولیه به متغیرها
- ۸- ثابت‌ها
- ۹- عملگر ورودی
- ۱۰- عملگرهای جایگزینی محاسباتی



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## مقدمه

- در دهه ۱۹۷۰ در آزمایشگاه‌های بل زبانی به نام C ایجاد شد.
- هنگامی که بحث شی‌گرایی و مزایای آن در جهان نرم‌افزار رونق یافت، زبان C که قابلیت شی‌گرایی نداشت ناقص به نظر می‌رسید تا این که در اوایل دهه ۱۹۸۰ دوباره شرکت بل دست به کار شد و زبان C++ را طراحی نمود.
- در سال ۱۹۹۸ زبان C++ توسط موسسه استانداردهای ملی آمریکا (ANSI) به شکل استاندارد و یک پارچه درآمد.

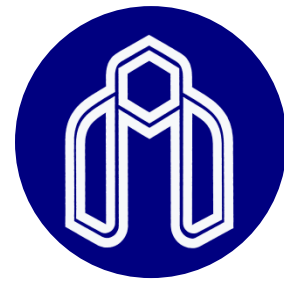


دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## آماده سازی مقدمات

□ یک «برنامه» دستورالعمل های متوالی است که می تواند توسط یک رایانه اجرا شود. برای نوشتن و اجرای هر برنامه به یک «ویرایش گر متن» و یک «کامپایلر» احتیاج داریم.

□ بسته Visual C++ محصول شرکت میکروسافت و بسته C++ Builder محصول شرکت بورلند نمونه های جالبی از محیط مجتمع تولید برای زبان C++ به شمار می روند.



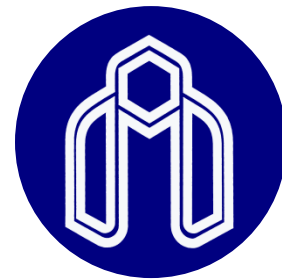
دانشگاه صنعتی شاهرود  
مرکز آموزش پای اکترونیکی

## شروع کار با C++

مثال : اولین برنامه

اولین برنامه‌ای که می‌نویسیم به محض تولد، به شما سلام می‌کند و عبارت "Hello, my programmer!" را نمایش می‌دهد:

```
#include <iostream>
int main()
{
    cout << "Hello, my programmer!\n" ;
    return 0;
}
```



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## شروع کار با ++C

اولین خط از کد بالا یک «راهنمای پیش‌پردازنده» است. راهنمای پیش‌پردازنده شامل اجزای زیر است:

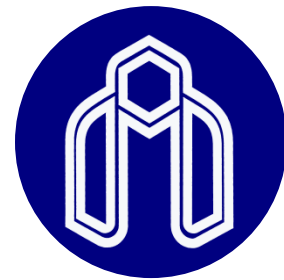
```
#include <iostream>
```

1- کاراکتر # که نشان می‌دهد این خط، یک راهنمای پیش‌پردازنده است.

این کاراکتر باید در ابتدای همه خطوط راهنمای پیش‌پردازنده باشد.

2- عبارت **include**

3- نام یک «فایل کتابخانه‌ای» که میان دو علامت < > محصور شده است.



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## شروع کار با ++C

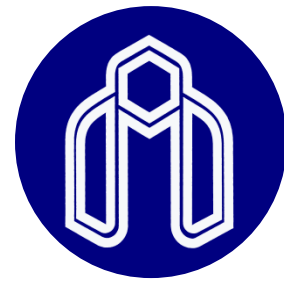
خط دوم برنامه نیز باید در همه برنامه‌های ++C وجود داشته باشد.

```
int main()
```

این خط به کامپایلر می‌گوید که «بدنه اصلی برنامه» از کجا شروع می‌شود. این خط دارای اجزای زیر است:

- ۱ - عبارت `int` که یک نوع عددی در ++C است.
- ۲ - عبارت `main` که به آن «تابع اصلی» در ++C می‌گویند.
- ۳ - دو پرانتز `()` که نشان می‌دهد عبارت `main` یک «تابع» است.

هر برنامه فقط باید یک تابع `main()` داشته باشد.



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## شروع کار با ++C

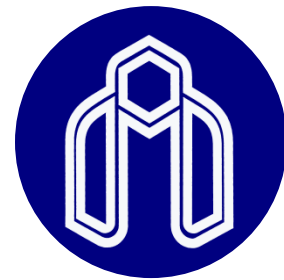
دستورات مابین { }، «بدنه اصلی برنامه» را تشکیل می دهند.

هر دستور با علامت سمیکولن ; پایان می یابد.

```
return 0;
```

آخرین دستور در اغلب برنامه ها





دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## شروع کار با ++C

### توضیح

توضیح، متنی است که به منظور راهنمایی و درک بهتر به برنامه اضافه می‌شود و تاثیری در اجرای برنامه ندارد. کامپایلر توضیحات برنامه را قبل از اجرا حذف می‌کند.

استفاده از توضیح سبب می‌شود که سایر افراد کد برنامه شما را راحت‌تر درک کنند.



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## شروع کار با ++C

به دو صورت می توانیم به برنامه های ++C توضیحات اضافه کنیم:

1 – با استفاده از دو علامت اسلش // :

هر متنی که بعد از دو علامت اسلش بیاید تا پایان همان سطر یک توضیح تلقی می شود .

2 – با استفاده از حالت C :

هر متنی که با علامت /\* شروع شود و با علامت \*/ پایان یابد یک توضیح تلقی می شود .



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

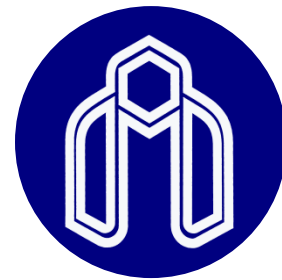
## عملگر خروجی

علامت << عملگر خروجی در C++ نام دارد (به آن عملگر درج نیز می‌گویند).

یک «عملگر» چیزی است که عملیاتی را روی یک یا چند شی انجام می‌دهد. عملگر خروجی، مقادیر موجود در سمت راستش را به خروجی سمت چپش می‌فرستد. به این ترتیب دستور

```
cout<< 66 ;
```

مقدار 66 را به خروجی cout می‌فرستد که cout معمولاً به صفحه‌نمایش اشاره دارد. در نتیجه مقدار 66 روی صفحه نمایش درج می‌شود.



دانشگاه صنعتی شاهرود  
مرکز آموزش پای‌الکترونیک

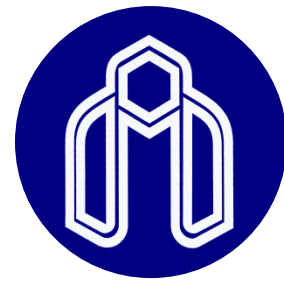
## لیترال‌ها و کاراکترها

یک «**لیترال**» رشته‌ای از حروف، ارقام یا علائم چاپی است که میان دو علامت نقل قول " " محصور شده باشد.

یک «**کاراکتر**» یک حرف، رقم یا علامت قابل چاپ است که میان دو نشانه ' ' محصور شده باشد. پس 'w' و '!' و '1' هر کدام یک کاراکتر است.

به تفاوت سه موجودیت «عدد» و «کاراکتر» و «لیترال رشته‌ای» دقت کنید: 6 یک عدد است، '6' یک کاراکتر است و "6" یک لیترال رشته‌ای است.

```
cout << a;  
cout << 'a';
```



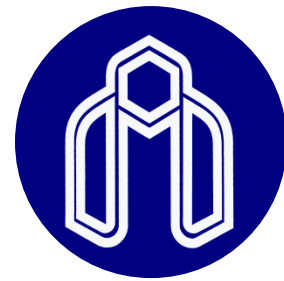
دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## کاراکترهای مخصوص

<code>\n</code>	<b>Newline</b>
<code>\t</code>	<b>Tab</b>
<code>\b</code>	<b>Backspace</b>
<code>\a</code>	<b>Beep sound</b>
<code>\"</code>	<b>Double quote</b>
<code>\'</code>	<b>Single quote</b>
<code>\0</code>	<b>Null character</b>
<code>\?</code>	<b>Question mark</b>
<code>\\</code>	<b>Back slash</b>

بعنوان مثال از کاراکتر `\a` می توان برای ایجاد صدای beep استفاده نمود.

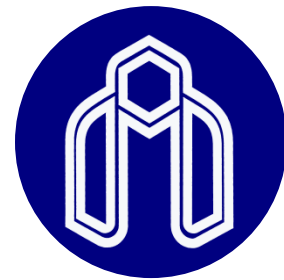
```
char x = '\a' ;  
cout << x;
```



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## متغیرها و تعریف آن‌ها:

«متغیر» مکانی در حافظه است که چهار مشخصه دارد: **نام**، **نوع**، **مقدار**،  
**آدرس**. وقتی متغیری را تعریف می‌کنیم، ابتدا با توجه به نوع متغیر، آدرسی از  
حافظه در نظر گرفته می‌شود، سپس به آن آدرس یک نام تعلق می‌گیرد.



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

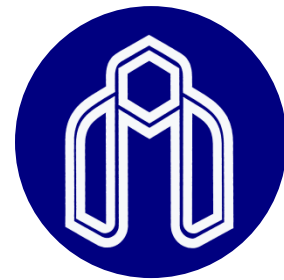
## متغیرها و تعریف آن‌ها:

در C++ قبل از این که بتوانیم از متغیری استفاده کنیم، باید آن را اعلان نماییم.

### نحوه‌ی اعلان یک متغیر

**type name =initializer ;**

عبارت **type** نوع متغیر را مشخص می‌کند. نوع متغیر به کامپایلر اطلاع می‌دهد که این متغیر چه مقادیری می‌تواند داشته باشد و چه اعمالی می‌توان روی آن انجام داد.



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## متغیرها و تعریف آن‌ها:

عبارت name نام متغیر را نشان می‌دهد. این نام حداکثر می‌تواند 31 کاراکتر باشد، نباید با عدد شروع شود، علائم ریاضی نداشته باشد و همچنین «کلمه کلیدی» نیز نباشد.

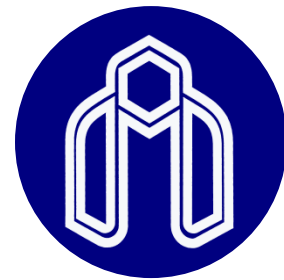
### مقداردهی اولیه

عبارت initializer عبارت «مقداردهی اولیه» نام دارد. با استفاده از این عبارت می‌توان مقدار اولیه‌ای در متغیر مورد نظر قرار داد.

دستور زیر تعریف یک متغیر صحیح را نشان می‌دهد:

```
int n = 50;
```





دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## مقداردهی اولیه به متغیرها

در بسیاری از موارد بهتر است متغیرها را در همان محلی که اعلان می‌شوند مقداردهی کنیم. استفاده از متغیرهای مقداردهی نشده ممکن است باعث ایجاد در دسرهایی شود.

مثلا اگر  $x$  را که مقداردهی نشده در عبارت  $y = x + 5$ ; به کار ببریم، حاصل  $y$  غیر قابل پیش‌بینی خواهد بود. برای اجتناب از چنین مشکلاتی عاقلانه است که متغیرها را همیشه هنگام تعریف، مقداردهی کنیم.

مثال:

```
int x=45;
```

```
int y=0;
```

```
int z;
```

```
z=0;
```



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## ثابت‌ها

در بعضی از برنامه‌ها از متغیری استفاده می‌کنیم که فقط یک بار لازم است آن را مقدار دهی کنیم و سپس مقدار آن متغیر در سراسر برنامه بدون تغییر باقی می‌ماند.

مثلا در یک برنامه محاسبات ریاضی، متغیری به نام PI تعریف می‌کنیم و آن را با 3.14 مقدار دهی می‌کنیم و می‌خواهیم که مقدار این متغیر در سراسر برنامه ثابت بماند. در چنین حالاتی از «**ثابت‌ها**» استفاده می‌کنیم.

یک **ثابت**، یک نوع **متغیر** است که فقط یک بار مقدار دهی می‌شود و سپس تغییر دادن مقدار آن در ادامه برنامه ممکن نیست.

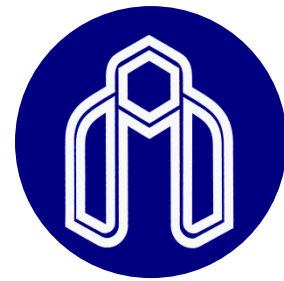
تعریف ثابت‌ها مانند تعریف متغیرهاست با این تفاوت که کلمه کلیدی **const** به ابتدای تعریف اضافه می‌شود.



## مثال تعریف ثابت‌ها:

```
int main()  
{ // defines constants; has no output:  
  const char BEEP = '\a';  
  const int MAXINT=2147483647;  
  const float DEGREE=23.53;  
  const double PI=3.14159265358979323846;  
  return 0;  
}
```

برنامه فوق خروجی ندارد:



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

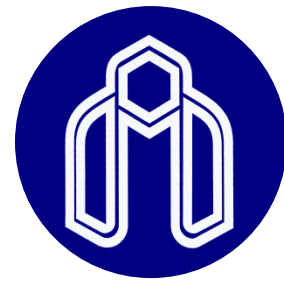
## عملگر ورودی

برای این که بتوانیم هنگام اجرای برنامه مقادیری را وارد کنیم از عملگر ورودی >> استفاده می کنیم.

استفاده از دستور ورودی به شکل زیر است:

```
cin >> variable;
```

*variable* نام یک متغیر است.



دانشگاه صنعتی شاهرود  
مرکز آموزش پای اکترودینی

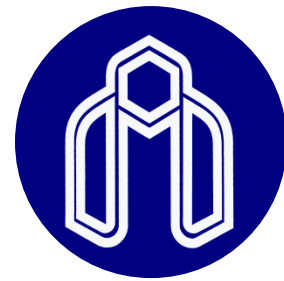
## عملگر ورودی

مثال: استفاده از عملگر ورودی

برنامه زیر یک عدد از کاربر گرفته و همان عدد را دوباره در خروجی نمایش می دهد:

```
#include<iostream>
int main()
{ // reads an integer from input:
  int m;
  cout << "Enter a number: ";
  cin >> m;
  cout << "your number is: " << m << endl;
  return 0;
}
```

```
Enter a number: 52
your number is: 52
```



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

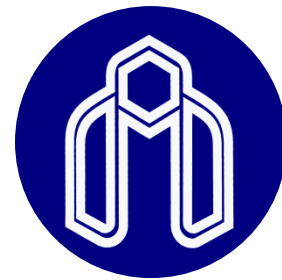
## عملگر ورودی

عملگرهای ورودی و خروجی به شکل جریانی رفتار می کنند.

مثلا با استفاده از دستور:

```
cin >> x >> y >> z;
```

سه مقدار **x** و **y** و **z** به ترتیب از ورودی دریافت می شوند. برای این کار باید بین هر ورودی یک فضای خالی (**space**) بگذارید و پس از تایپ کردن همه ورودی ها، کلید **enter** را بفشارید.



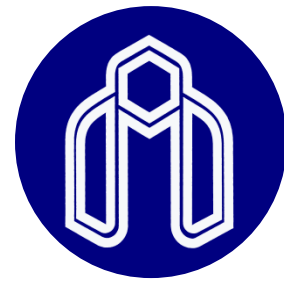
دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## عملگر انتساب / عملگرهای جایگزینی محاسباتی

عملگر انتساب = می باشد که باعث می گردد مقدار عبارت در طرف راست این عملگر ارزیابی شده و در متغیر طرف چپ آن قرار گیرد.

از ترکیب عملگرهای محاسباتی و عملگر (=)، عملگرهای جایگزینی محاسباتی بوجود می آیند. این عملگرها اعمال محاسباتی و انتساب را انجام می دهند.

نام عملگر	علامت	مثال	عملکرد
انتساب جمع	$+=$	$a += 1$	$a = a + 1$
انتساب تفریق	$-=$	$a -= 1$	$a = a - 1$
انتساب ضرب	$*=$	$a *= 2$	$a = a * 2$
انتساب تقسیم	$/=$	$a /= 2$	$a = a / 2$
انتساب باقی مانده تقسیم	$\%=$	$a \% = 2$	$a = a \% 2$



دانشگاه صنعتی شاهرود  
مرکز آموزش های الکترونیکی

## مثال

مثال: دو عدد از نوع اعشاری را گرفته مجموع و حاصلضرب آنها را محاسبه و نمایش می‌دهد.

```
#include <iostream>
int main()
{
float    x,y,s,p ;
cin >> x >> y ;
s= x+y ;
p=x*y;
cout << s <<endl << p;
return 0 ;
}
```





۱- برنامه ای بنویسید که شعاع یک دایره را از ورودی خوانده و محیط و مساحت آن را در خروجی چاپ کند.

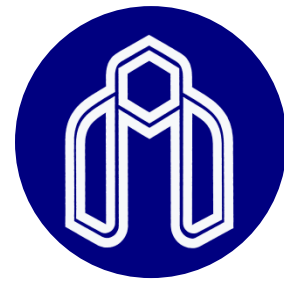
۲- برنامه ای بنویسید که مقدار X را از ورودی خوانده و مقدار Y را در خروجی چاپ کند:

$$y=1/(x^2+x+3)$$

۳- برنامه ای بنویسید که خروجی زیر را تولید کند:

The answer to the question of  
Life, the universe, and Everything is 42.

۴- برنامه ای بنویسید که وزن کالایی را برحسب کیلوگرم دریافت کرده، وزن آن را برحسب گرم در خروجی چاپ کند.



با تشکر از توجه شما