

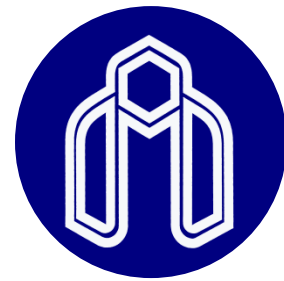
مبانی کامپیوتر و برنامه نویسی



جلسه چهارم: مقدمات برنامه نویسی با ++C

ارائه دهنده: یگانه جلالی

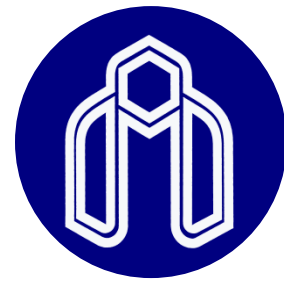




دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

{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



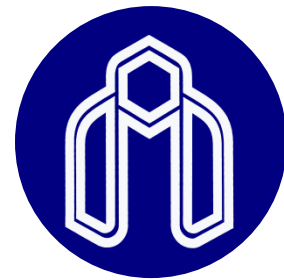
دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

مقدمه بر برنامه‌نویسی C++

📅 زبان C++ در دهه ۱۹۸۰ توسط بیارنه استراستروپ طراحی و توسعه داده شد.

📅 این زبان از زبان C به ارث برده است، ولی امکانات جدیدی مانند شیء‌گرایی، ارث‌بری و مدیریت دستی حافظه را به آن اضافه کرده است.

📅 در برنامه‌نویسی سیستم‌ها، بازی‌ها، نرم‌افزارهای پیچیده و حتی برنامه‌های کاربردی با عملکرد بالا استفاده می‌شود.



دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

ویژگی های زبان ++C

پشتیبانی از کتابخانه غنی ۴

سرعت بالا ۵

کامپایلری ۶

سطح میانی ۱

شی گرا ۲

ساخت یافته ۳

سیستم های مدیریت پایگاه داده

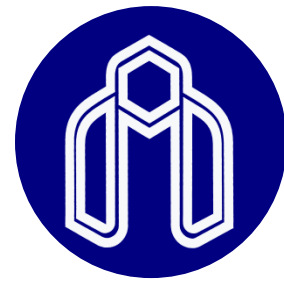


صنعت بازی سازی



سیستم عامل ها و نرم افزارهای کاربردی

++C در دنیای واقعی

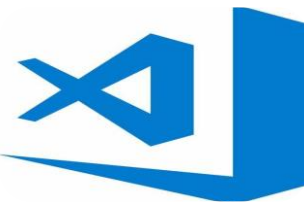


دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

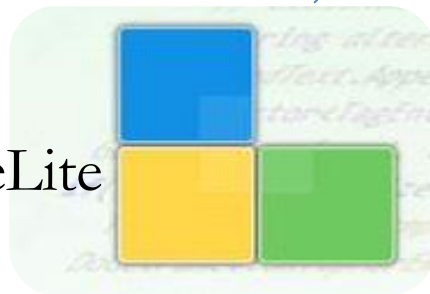
محیط توسعه یکپارچه (IDE)

یک «برنامه» دستورالعمل‌های متوالی است که می‌تواند توسط یک رایانه اجرا شود. برای نوشتن و اجرای هر برنامه به یک «ویرایش‌گر متن» و یک «کامپایلر» احتیاج داریم.

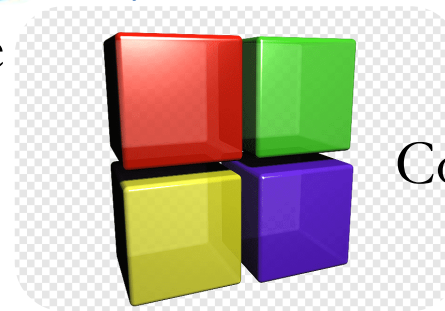
چند نمونه IDE معروف برای ++C:



VS code



codeLite



Codeblocks

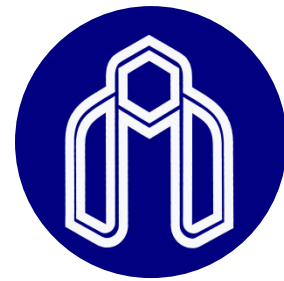
eclipse



eclipse



مفد CLion



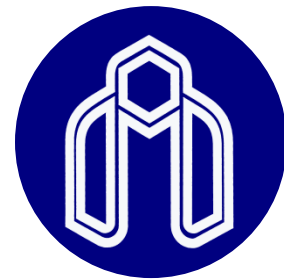
دانشگاه صنعتی شاهرود
مرکز آموزش پای اکتردیگی

شروع کار با C++

مثال : اولین برنامه

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

```
#include <iostream>
int main()
{
    cout << "Hello, world!" << endl ;
    return 0;
}
```



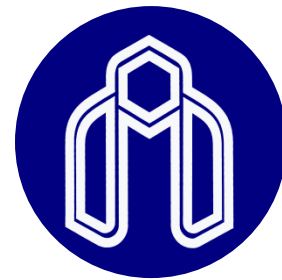
دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

شروع کار با ++C

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

```
#include <iostream>
```

- ۱- کاراکتر # که نشان می دهد این خط، یک راهنمای پیش پردازنده است. این کاراکتر باید در ابتدای همه خطوط راهنمای پیش پردازنده باشد.
- ۲- عبارت `include`
- ۳- نام یک «فایل کتابخانه ای» که میان دو علامت `<>` محصور شده است.



دانشگاه صنعتی شاهرود
مرکز آموزش پای اکترونیکی

شروع کار با ++C

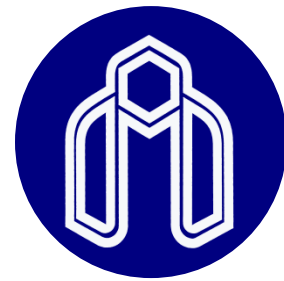
خط دوم برنامه نیز باید در همه برنامه‌های ++C وجود داشته باشد.

```
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

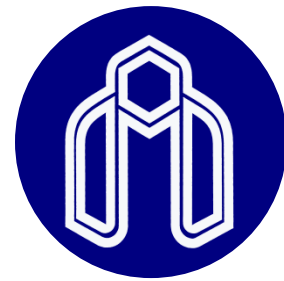
شروع کار با ++C

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



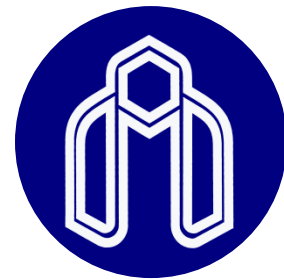
دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

شروع کار با ++C

توضیح

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

استفاده از توضیح سبب می شود که سایر افراد کد برنامه شما را راحت تر درک کنند.



دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

شروع کار با ++C

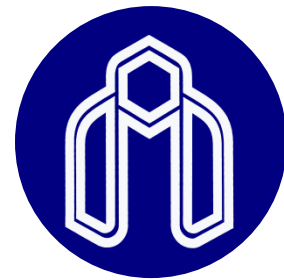
به دو صورت می توانیم به برنامه های ++C توضیحات اضافه کنیم:

۱ - با استفاده از دو علامت اسلش // :

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

۲ - 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

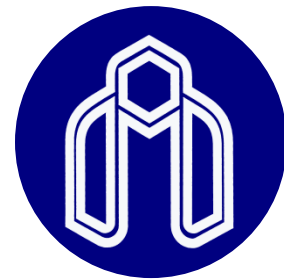
علامت << عملگر خروجی در C++ نام دارد (به آن عملگر درج نیز می گویند).

یک «عملگر» چیزی است که عملیاتی را روی یک یا چند شی انجام می دهد. عملگر خروجی، مقادیر موجود در سمت راستش را به خروجی سمت چپش می فرستد.

به این ترتیب دستور

```
cout<< 66 ;
```

مقدار 66 را به خروجی cout می فرستد که cout معمولاً به صفحه نمایش اشاره دارد. در نتیجه مقدار 66 روی صفحه نمایش درج می شود.



دانشگاه صنعتی شاهرود
مرکز آموزش پای الکترونیک

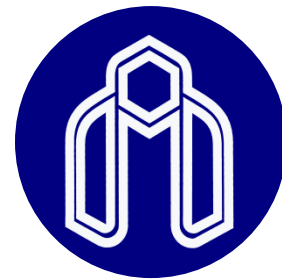
لیترال‌ها و کاراکترها

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';
```



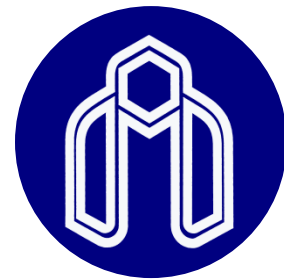
دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

کاراکترهای مخصوص

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

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

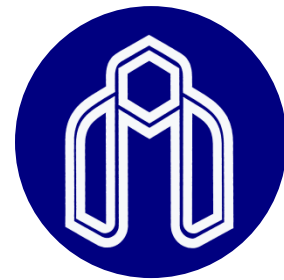
```
char x = '\a' ;  
cout << x;
```



دانشگاه صنعتی شاهرود
مرکز آموزش پای اکترونیکی

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

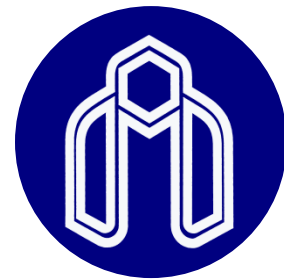
متغیرها و تعریف آن‌ها:

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:

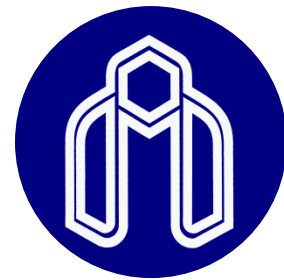
عبارت `name` نام متغیر را نشان می‌دهد. این نام حداکثر می‌تواند ۳۱ کاراکتر باشد، نباید با عدد شروع شود، علائم ریاضی نداشته باشد و همچنین «کلمه کلیدی» نیز نباشد.

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 پای اکترود

مقداردهی اولیه به متغیرها

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

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

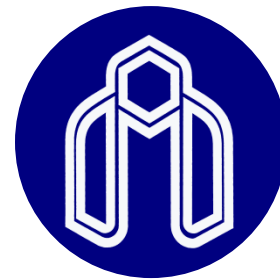
مثال:

```
int x=45;
```

```
int y=0;
```

```
int z;
```

```
z=0;
```



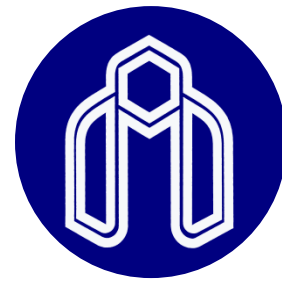
دانشگاه صنعتی شاهرود
مرکز آموزش پای الکترونیکی

ثابت‌ها

در بعضی از برنامه‌ها از متغیری استفاده می‌کنیم که فقط یک بار لازم است آن را مقداردهی 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 مقداردهی 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 مقداردهی 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.



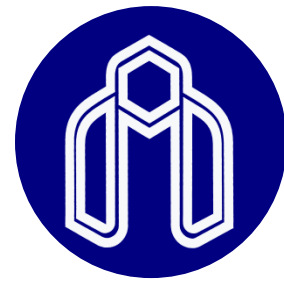
دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

ثابت‌ها

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 پای اکترودنی

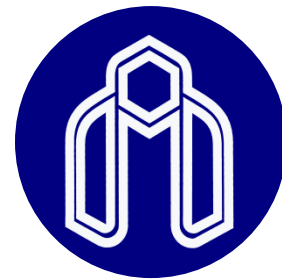
عملگر ورودی

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.



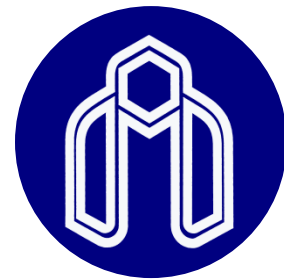
دانشگاه صنعتی شاهرود
مرکز آموزش های الکترونیکی

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
```



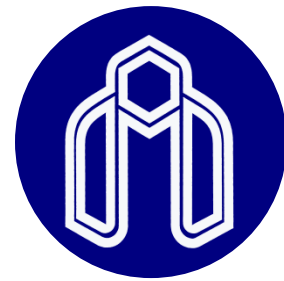
دانشگاه صنعتی شاهرود
مرکز آموزش پای الکترونیک

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.

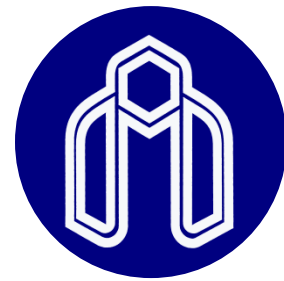


عملگر انتساب / عملگرهای جایگزینی محاسباتی

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	$\%=$	$a\%= 2$	$a=a\%2$

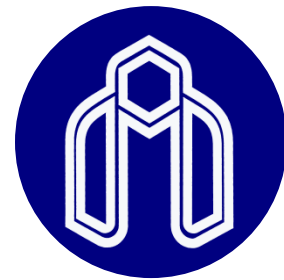


دانشگاه صنعتی شاهرود
مرکز آموزش پای اکترودینی

مثال

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

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



با تشکر از توجه شما

ارائه دهنده: یگانه جلالی