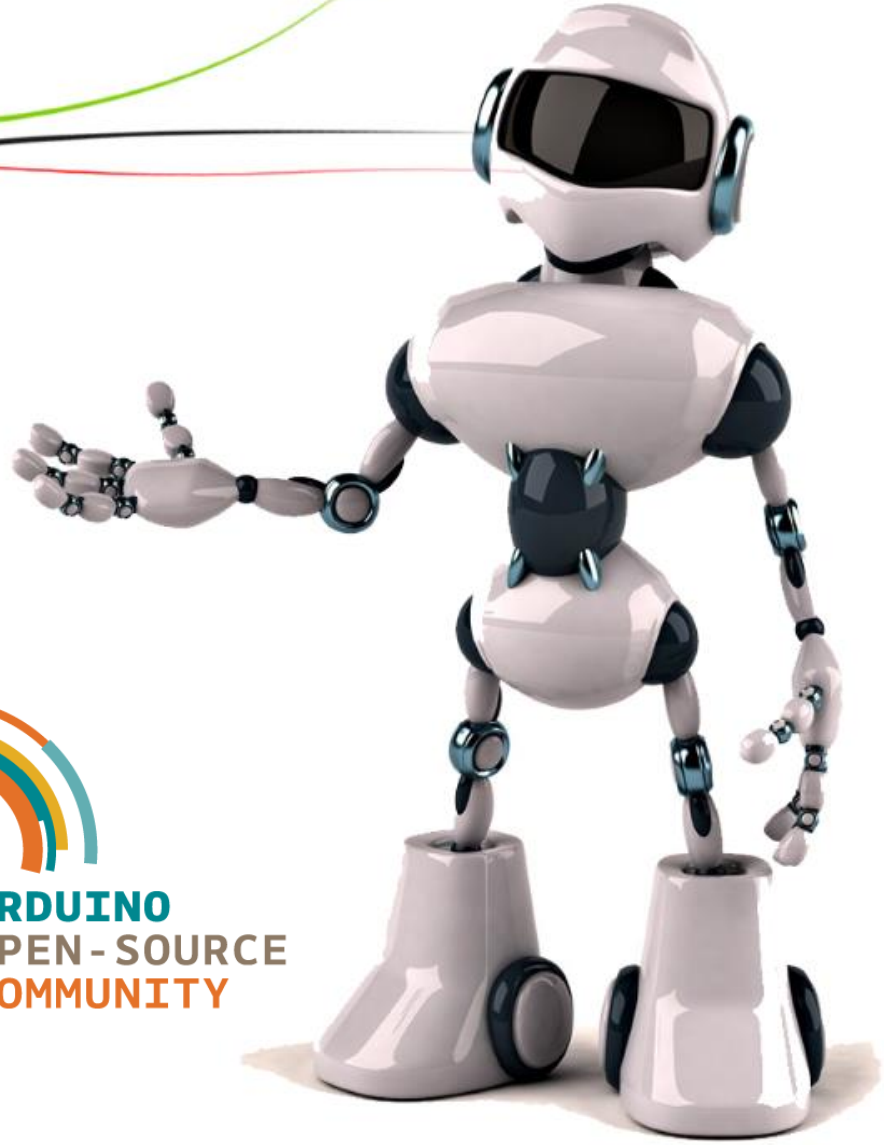
مکتب خونه
آکادمی آنلاین تخصص ها



ARDUINO
OPEN-SOURCE
COMMUNITY

دوره آموزشی آردوینو

فصل دوم

بخش ششم : عملگر های محاسباتی و مقایسه ای

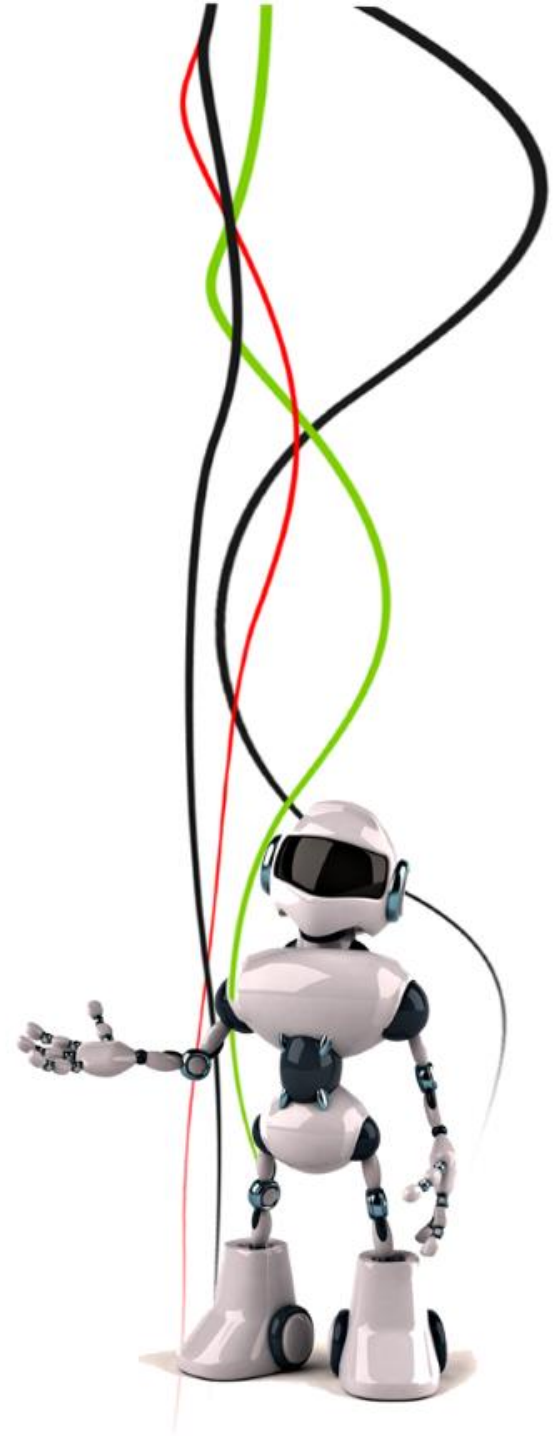
مدرس : محمد پارسا کریمی

عملگرها

- عملگر(operator): نمادی که عمل خاصی را انجام می‌دهد.
- عملوند (operand): مقادیری که عملگرها روی آن‌ها عمل می‌کنند.

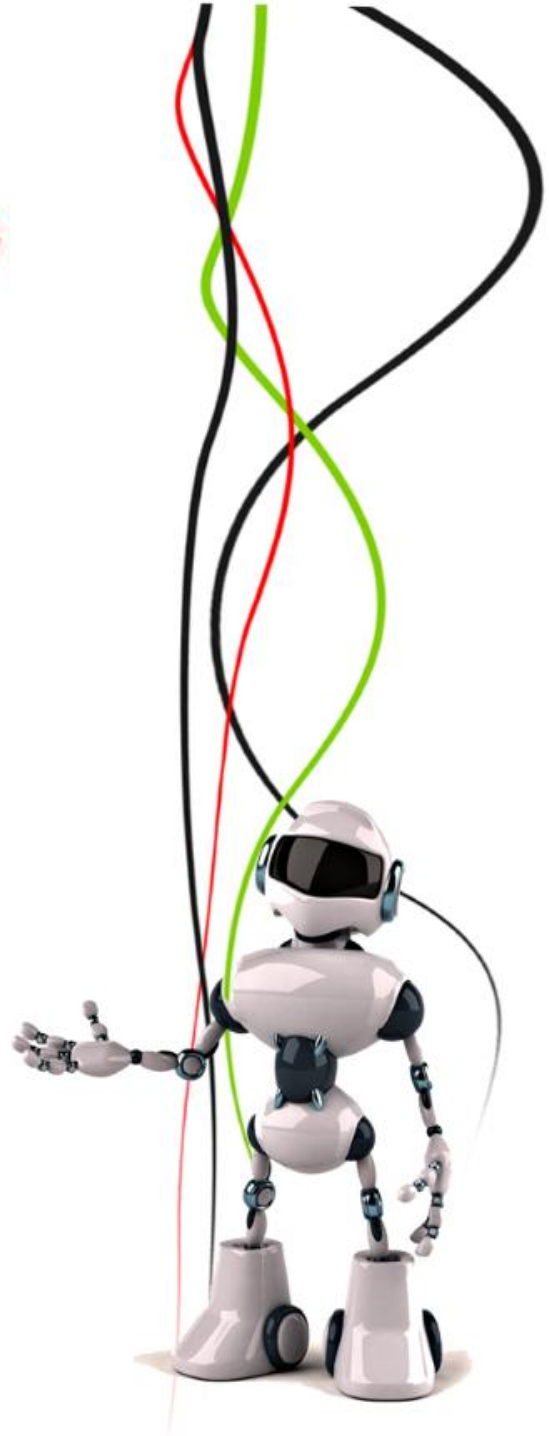
انواع عملگرها

1. عملگرهای محاسباتی +, %, -
2. عملگرهای رابطه‌ای < = >
3. عملگرهای منطقی &&, ||
4. عملگرهای بیتی &, |, ^



عملگرهای محاسباتی

عملگر	نام	مثال
-	تفریق و منهای یکانی	x - y یا -x
+	جمع	x + y
*	ضرب	x * y
/	تقسیم	x / y
%	باقیمانده تقسیم	x % y
--	کاهش (decrement)	-- x یا x --
++	افزایش (increment)	++ x یا x ++



نکات عملگرهای محاسباتی

- چهار عملگر + ,- ,* ,/ تقریباً روی همه انواع داده‌ای در C++ به‌کار می‌روند.
- چنانچه عملگر / روی مقادیر صحیح یا کاراکتر اعمال شود، قسمت اعشار در پاسخ حذف خواهد شد.

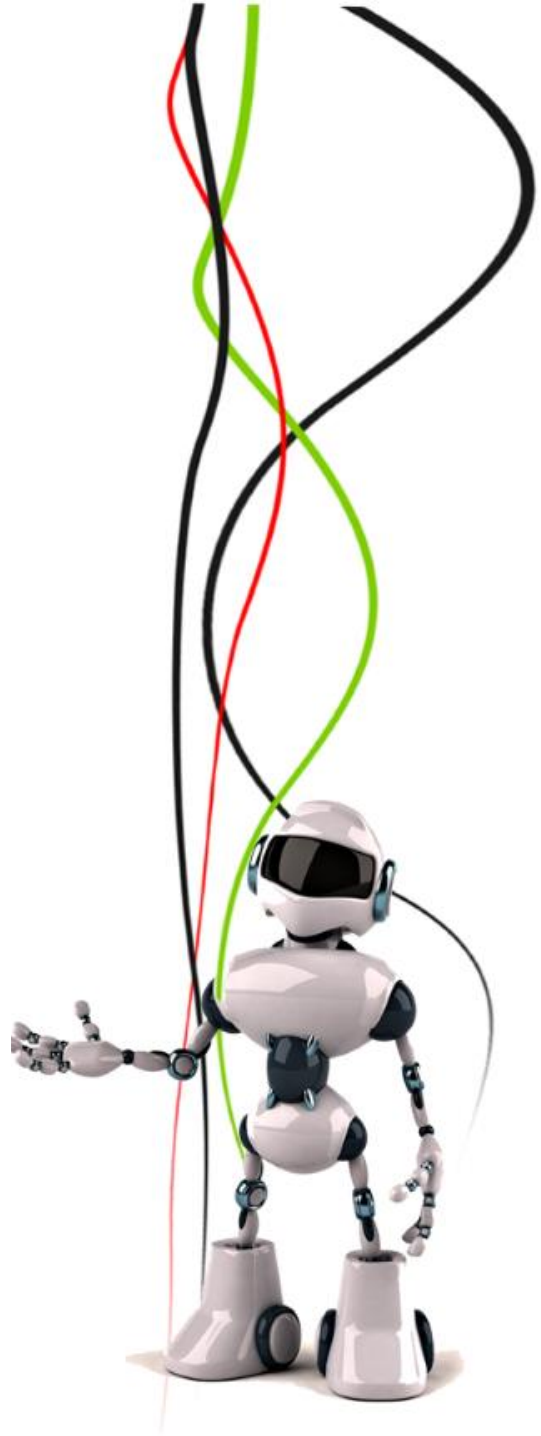
16/5=3

- هر دو عملوند برای عملگر % باید از نوع صحیح باشند.

```
int  x=10, y=4,z;
float  k=2.5;
z=x%y;
z=x%k;
```

z=2

ERROR

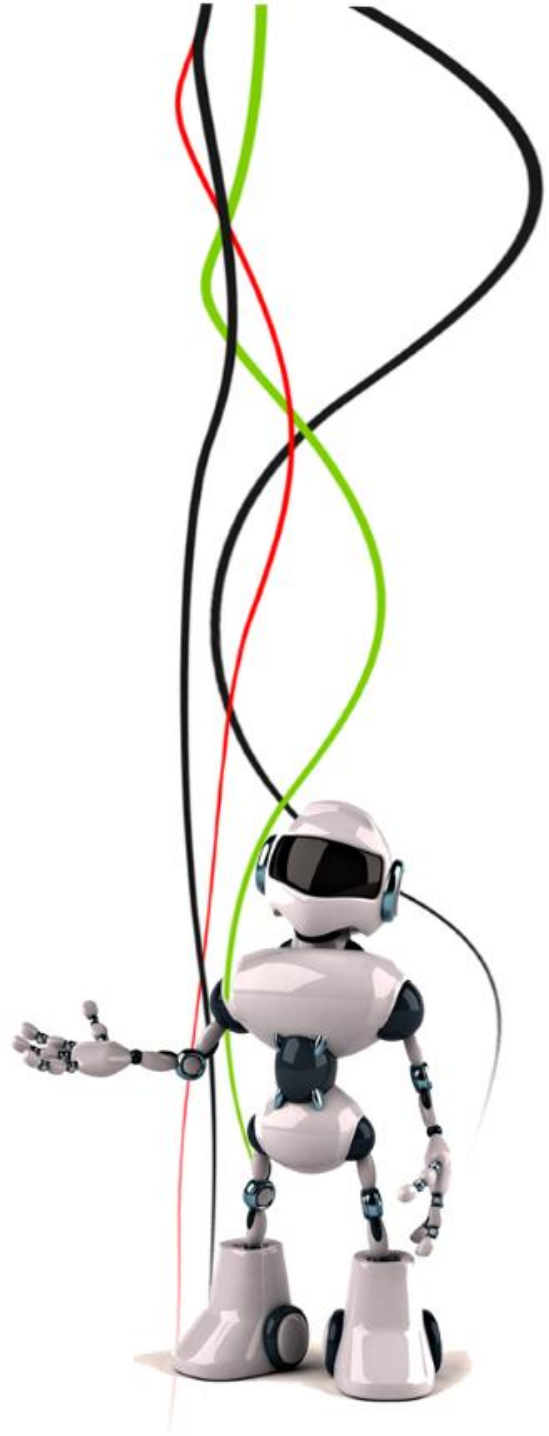


نکات عملگرهای محاسباتی

مثال

```
int a=10, b=3;
float C=12.5, d=2.0;
```

عبارت محاسباتی	مقدار	عبارت محاسباتی	مقدار
a + b	13	C + d	14.5
a - b	7	C - d	10.5
a * b	30	C * d	25.0
a / b	3	C / d	6.25
a % b	1	C % d	error

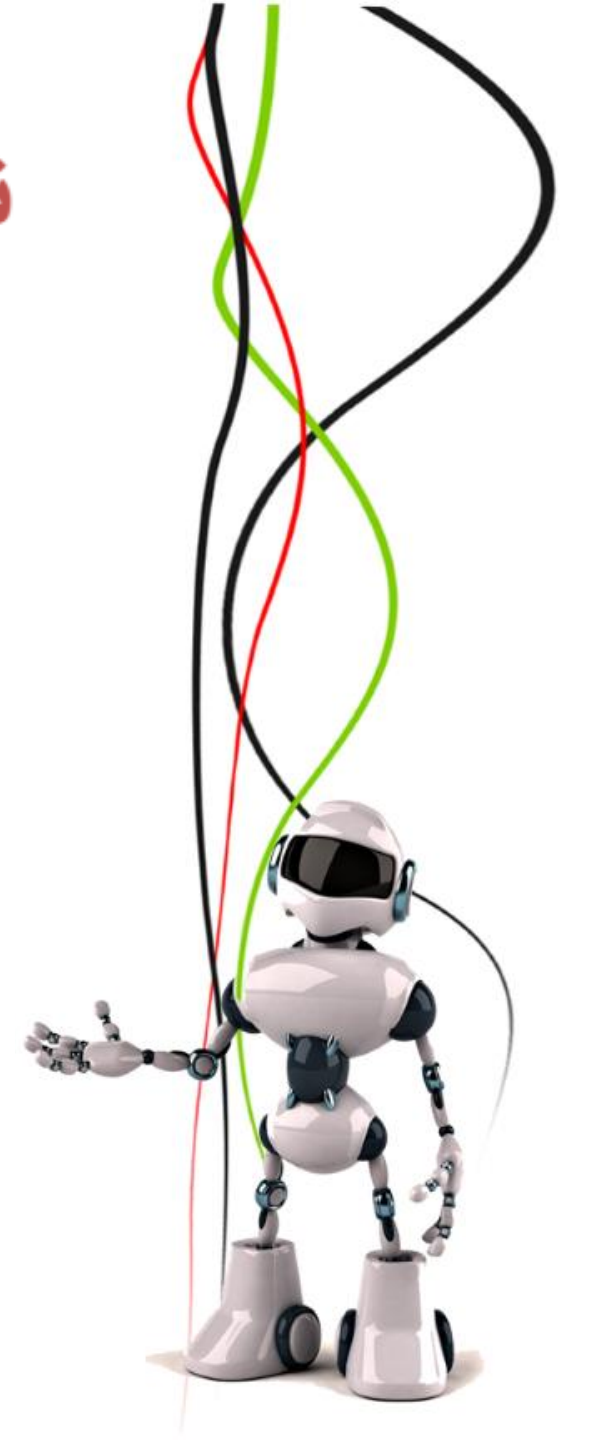


نکات عملگرهای محاسباتی

char c1='A', c2='E';

عبارت محاسباتی	مقدار
c1	65
c1 + c2	134
c1 + c2 + 5	139
c1 + c2 + '5'	187

- در عبارات فوق به جای c1 و c2 از کد اسکی آن‌ها (65 و 69) استفاده شده‌است.
- دقت کنید که عدد 5 با کاراکتر '5' که کد اسکی آن 53 است تفاوت دارد.

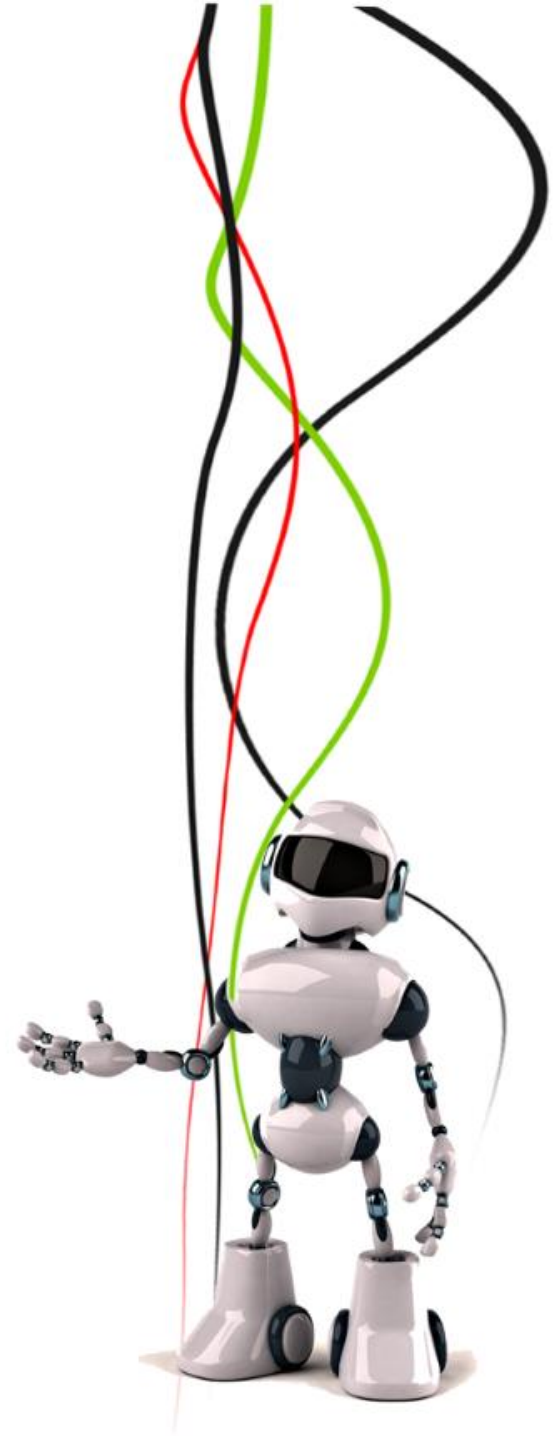


نکات عملگرهای محاسباتی

- اگر دو عملوند در عملگر انتساب (=) از نوع یکسان نباشند، مقدار عبارت یا عملوند سمت راست به طور خودکار به نوع متغیر سمت چپ تغییر خواهد یافت.

```
int a, b=5;
```

عبارت	مقدار a
a=b	
a=4.34	
a=b/2	
a='x'	



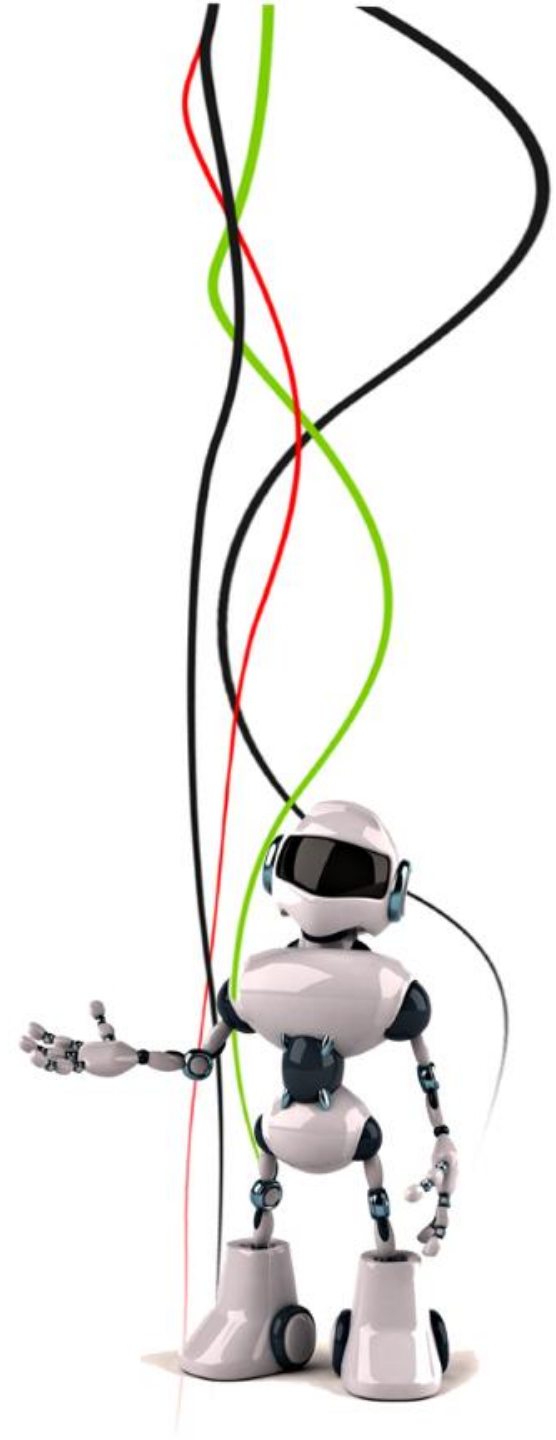
عملگرهای محاسباتی

- عملگر کاهش (--): یک واحد از عملوندش کم می‌کند و نتیجه را در همان عملوند قرار می‌دهد.
- عملگرافزایش (++): یک واحد به عملوندش اضافه می‌کند و نتیجه را در همان عملوند قرار می‌دهد.

```
int  a=8, b=3;
b++;  ⟶  b=4
--a;  ⟶  a=7
```

a- - و - -a معادل‌اند با a=a-1
a++ و ++a معادل‌اند با a=a+1

در دستوراتی که عملگرهای کاهش و افزایش به تنهایی استفاده می‌شوند
محل قرارگیری عملوند تاثیری در پاسخ ندارد. اما عملکرد آن‌ها در عبارات محاسباتی متفاوت می‌باشد.



عملگرهای محاسباتی

- در عبارات محاسباتی، چنانچه عملگرهای - - و++ قبل از عملوند قرار گیرند، ابتـدا عملگرها بر روی عملوند عمل کرده و نتیجه‌ی آن در محاسبات شرکت می‌کند.

```
int a, b;
a=5;
b=++a;
```

a=6

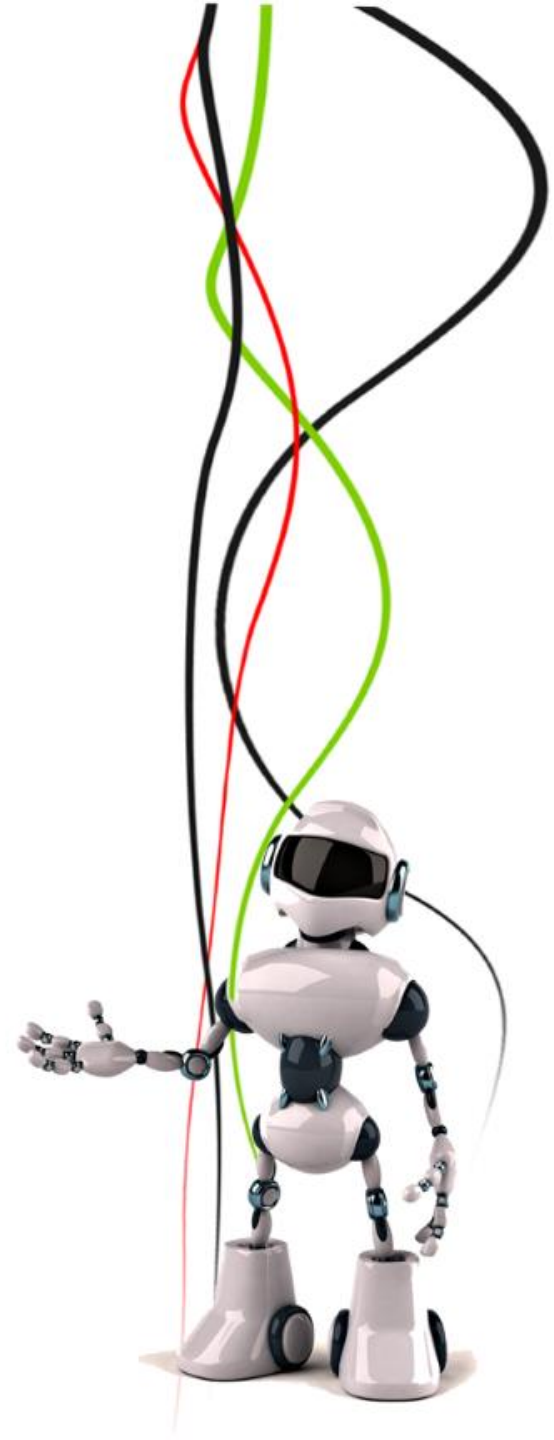
b=6

- در عبارات محاسباتی، چنانچه عملگرهای- - و++ بعد از عملوند قـرار گیرنـد، مقـدار فعلی عملوند در محاسبات شرکت کرده و نتیجه عبارت محاسبه می‌شـود و در نهایـت عملگر - - یا ++ بر روی عملوندش اعمال می‌شود.

```
int a, b;
a=5;
b=a++;
```

b=5

a=6



تقدم (اولویت) عملگرهای محاسباتی

++ --	بالاترین تقدم
-(منهای یکانی)	
* / %	
+ -	پایین‌ترین تقدم

عملگرهایی که دارای تقدم یکسان هستند، مانند *، / و ٪، در یک عبارت محاسباتی نسبت به هم تقدم مکانی دارند یعنی هر عملگری که زودتر ظاهر شود، زودتر نیز اجرا خواهد شد.

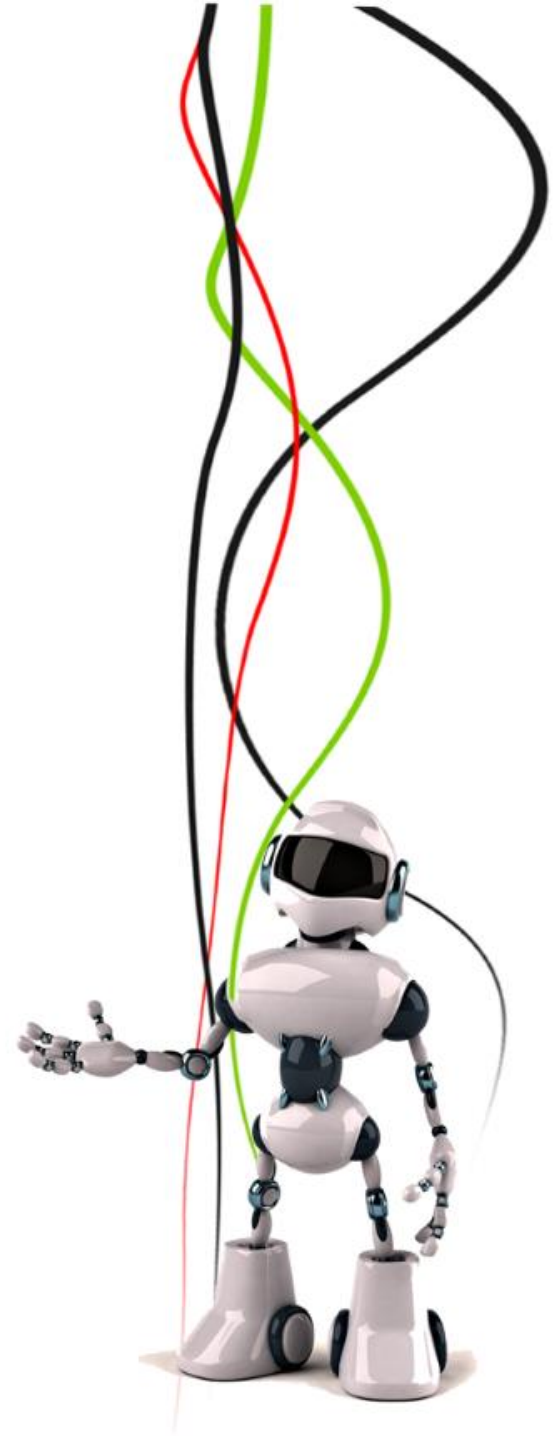
```
int x,y=6, z=10;
x=y+z/2*3;
```

3 1 2

5

15

21

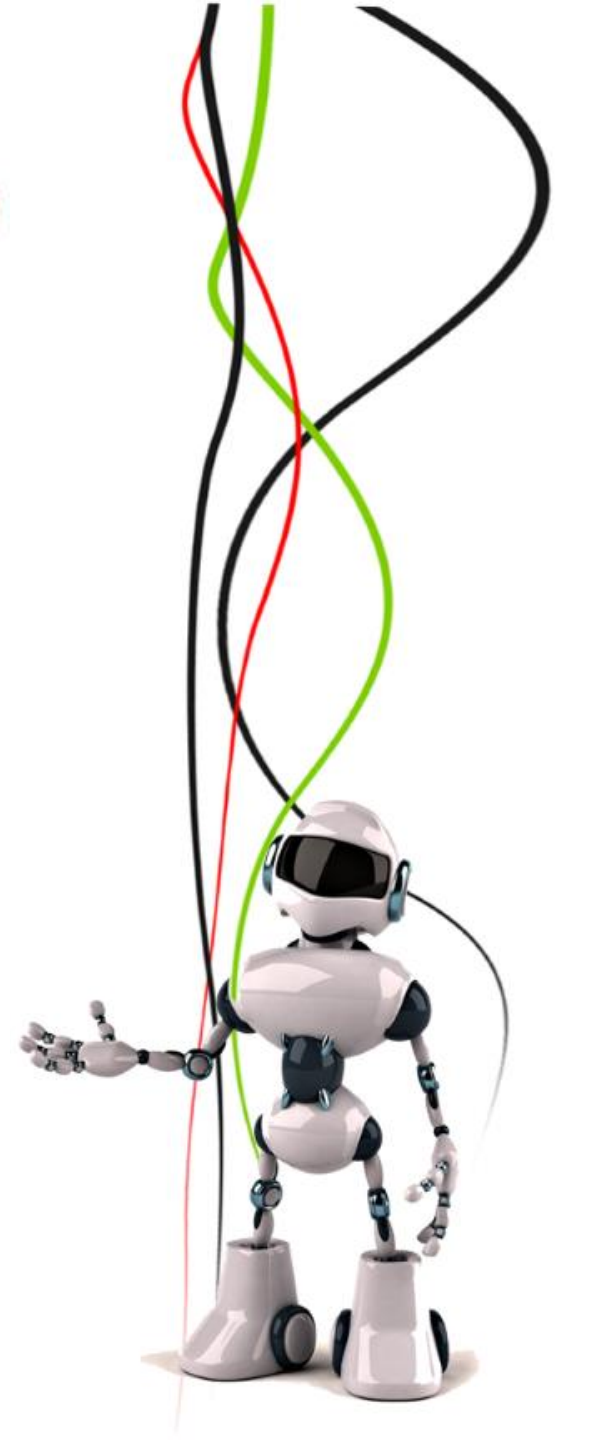


عملگرهای رابطه‌ای

عملگر	نام	مثال
>	بزرگتر	x > y
>=	بزرگتر یا مساوی	x >= y
<	کوچکتر	x < y
<=	کوچکتر یا مساوی	x <= y
==	متساوی	x == y
!=	نامساوی	x != y

عملگرهای == و =! برای مقایسه‌ی دو مقدار مورد استفاده قرار می‌گیرد.

معادل در C++	عبارت ریاضی
a+b<=35	a+b≤35
3*a-b!=2	3a-b≠2



مثال: جمع کردن اعداد صحیح

برنامه‌ای که جمع دوعدد صحیح دریافتی از کاربر را در خروجی چاپ کند.

```
#include <iostream>
using namespace std;
int main()
{
//variable declaration
int number1;
int number2;
int sum;

cout << "Enter first number: ";
cin>>number1;

cout << "Enter second number: ";
cin>>number2;

sum= number1+ number2;
cout << "Sum is"<< sum;

return 0;
} // end of function main
```

تعریف (اعلان) متغیرهای برنامه

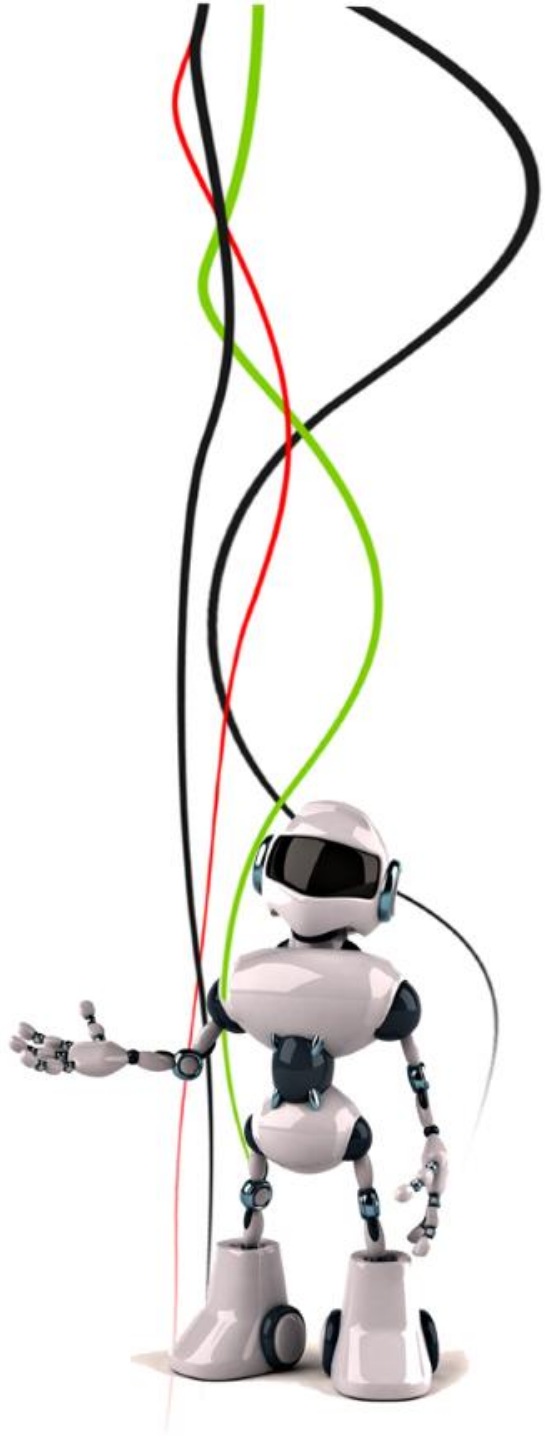
میتوان تعریف را به این شکل نوشت:
int number1, number2, sum;

با این دستور یک راهنمایی برای کاربر در خروجی چاپ می‌شود

با دستور cin و عملگر استخراج جریان << یک ورودی از صفحه کلید دریافت شده و در متغیر number1 قرار می‌گیرد.

این دستور مجموع مقادیر number1 و number2 را محاسبه کرده و با دستور انتساب به متغیر sum نسبت میدهد.

این دستور رشته کاراکتری Sum is و را به همراه مقدار عددی sum به خروجی میفرستد



نکات

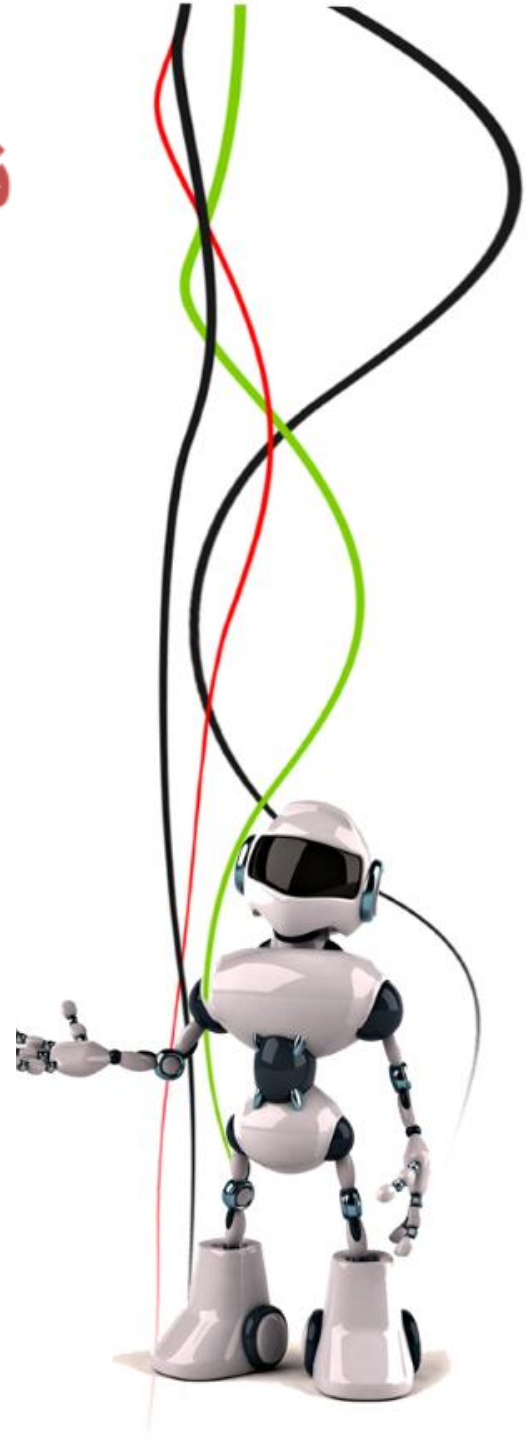
```
#include <iostream>
using namespace std;
int main()
{
//variable declaration
int number1;
int number2;
int sum;

cout << "Enter first number: ";
cin>>number1;

cout << "Enter second number: ";
cin>>number2;

sum= number1+ number2;
cout << "Sum is"<< sum;

return 0;
} // end of function main
```

اعلان متغیرها را تقریبا در هر نقطه‌ای از برنامه می‌توان قرار داد
اما باید آنها را قبل از اولین استفاده از متغیر موردنظر تعریف کنید.

هنگام اجرای دستور cin برنامه منتظر ورود یک داده از صفحه کلید میماند.
کاربر با وارد کردن ورودی موردنظر و سپس کلید Enter به این انتظار پایان میدهد.

چنانچه بخواهید یک عبارت توسط cout عینا در خروجی چاپ شود
آنرا در کوتیشن دوتایی قرار میدهیم. در غیر اینصورت مقدار متغیر
یا عبارت در خروجی چاپ میشود.

در دستورات خروجی میتوان محاسبات نیز انجام داد:
cout << "Sum is"<< number1+number2;
در اینصورت نیازی به متغیر sum نمیباشد.

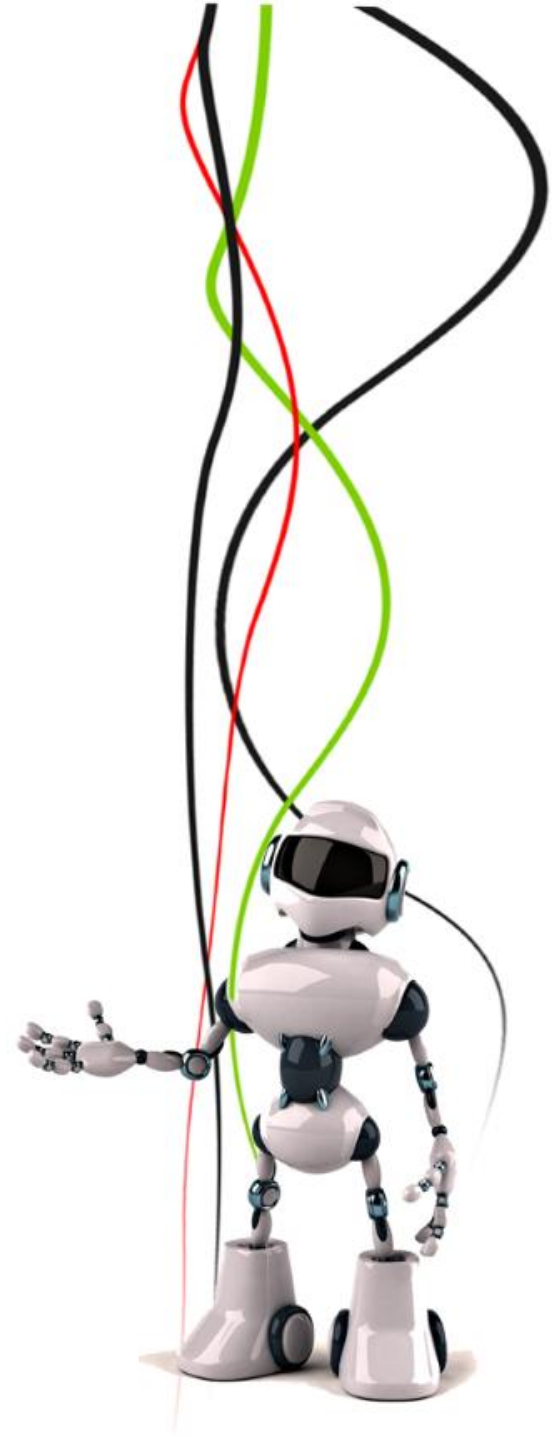


مثال: متغیر مقداردهی نشده

```
#include <iostream>
using namespace std;
int main()
{

  int x;
  int y=45;

  cout << "x = " << x << " and y = " << y << endl;
  return 0;
}
```

نکته مهم:
روش برخورد کامپایلرهای مختلف با متغیرهای مقداردهی نشده متفاوت است.
اغلب کامپایلرها مقداری که قبلا در آن قسمت از حافظه بوده و سپس بدون استفاده رها شده است را در متغیر قرار میدهند، برخی مقدار ؟ را جایگزین کرده و انواعی از کامپایلرها به طور پیش فرض مقدار ۰ را جایگزین میکنند.

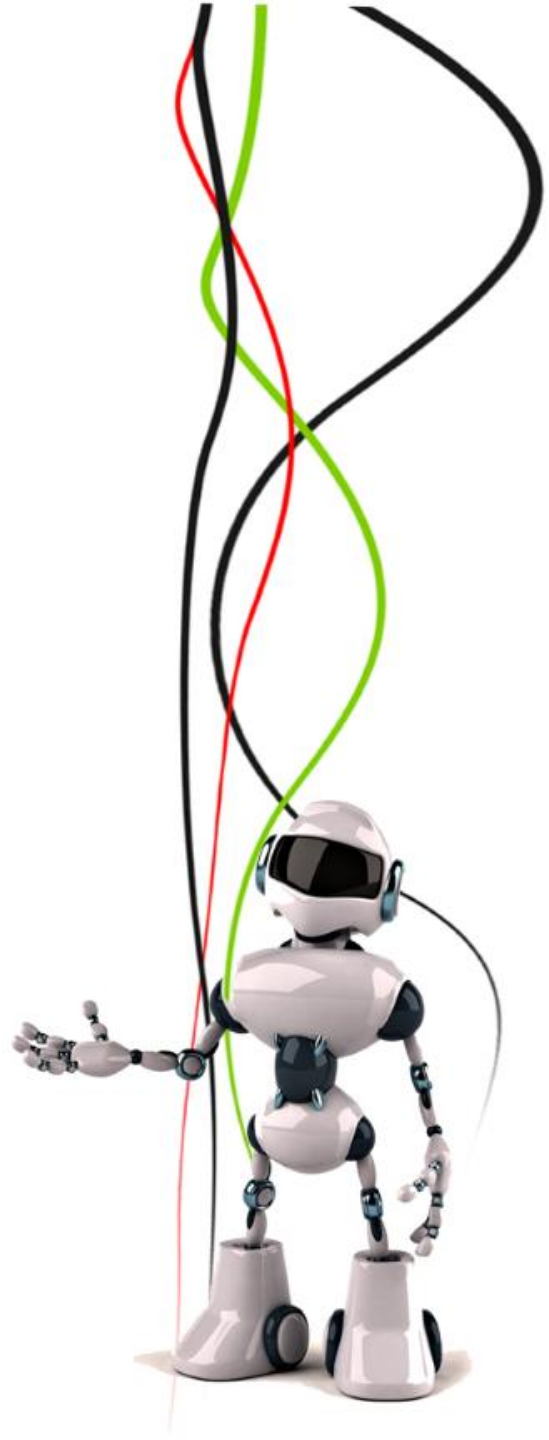
دستور endl شکل اختصاری end line است و به فضای نام std تعلق دارد.
این دستور یک خط جدید را به خروجی میفرستد و سپس بافر خروجی را تخلیه مینماید.



مشاهده‌ی صفحه‌ی خروجی

- معمولاً پس از اجرای برنامه و چاپ خروجی‌ها، کامپیوتر بلافاصله به برنامه برمی‌گردد و شما قادر به دیدن خروجی‌ها نخواهید بود.
- برای رفع این مشکل می‌بایست از دستور cin.get() در انتهای برنامه‌ها قبل از دستور return استفاده کنید.
- این تابع صفحه‌ی خروجی را تا زمانی که کلیدی را فشار ندهید، نمایش خواهد داد.

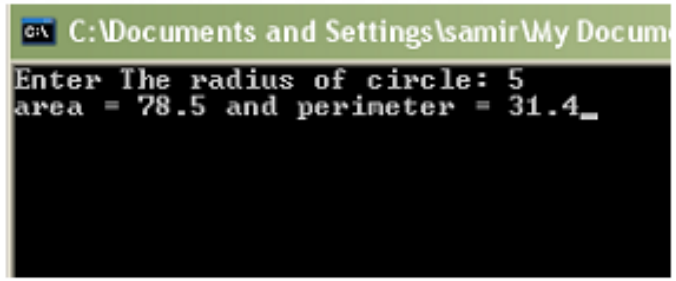
بهتر است، عادت کنید این تابع را به طور پیش‌فرض در تمامی برنامه‌های خود قرار دهید.

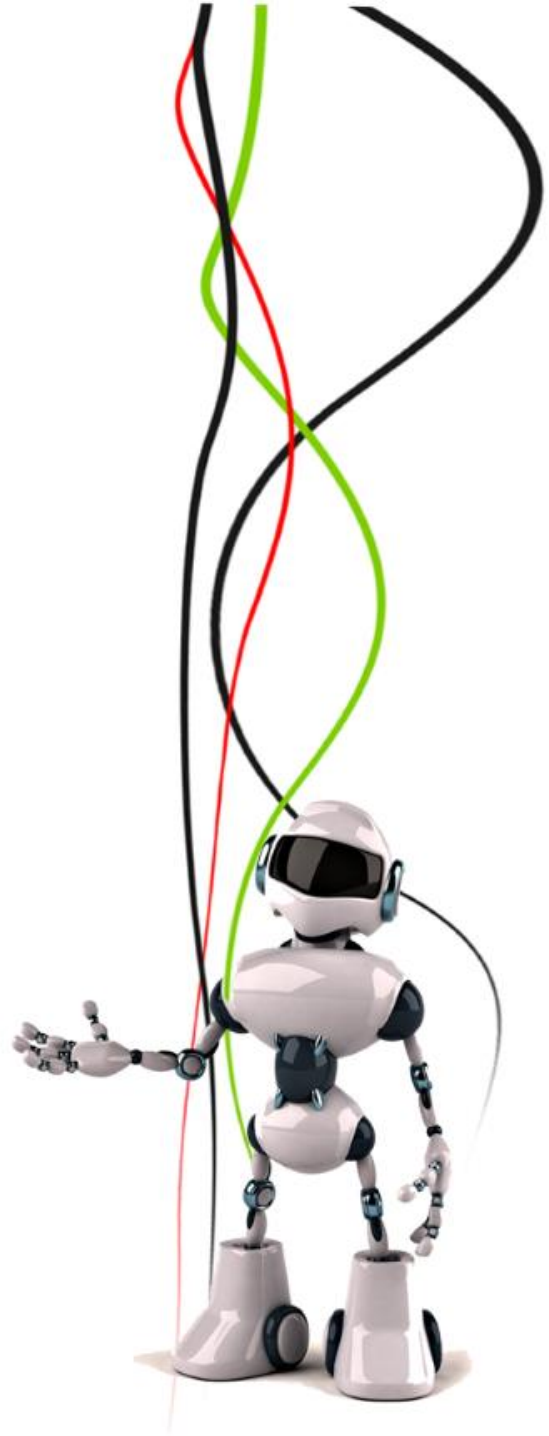


مثال: محاسبه محیط و مساحت دایره

```
#include <iostream>
#define  PI  3.14
using namespace std;
int main()
{
  int radius;
  float area, perimeter;

  cout<<"Enter The radius of circle: ";
  cin>> radius;
  area=PI * radius* radius;
  primeter=2* PI*radius;
  cout << "area = " << area << " and perimeter = " << perimeter;
  cin.get();
  cin.get();
  return 0;
}
```

یک روش دیگر برای تعریف مقادیر ثابت





علت استفاده از دو دستور cin.get()

- دقت کنید هنگامی که از دستور cin برای دریافت اطلاعات از صفحه کلید استفاده میکنید این دستور کد کلید Enter را در بافر صفحه کلید که محل نگهداری داده‌های ورودی است باقی می‌گذارد.
- به همین دلیل دستور بعدی cin با دیدن کد کلید enter آنرا خوانده و بلافاصله صفحه نمایش بسته می‌شود.
- بنابراین برای حذف این کد نیاز به یک دستور cin مجزا می‌باشد.

علت استفاده از دو دستور cin.get()

- دقت کنید هنگامی که از دستور cin برای دریافت اطلاعات از صفحه کلید استفاده میکنید این دستور کد کلید Enter را در بافر صفحه کلید که محل نگهداری داده‌های ورودی است باقی می‌گذارد.
- به همین دلیل دستور بعدی cin با دیدن کد کلید enter آنرا خوانده و بلافاصله صفحه نمایش بسته می‌شود.
- بنابراین برای حذف این کد نیاز به یک دستور cin مجزا می‌باشد.