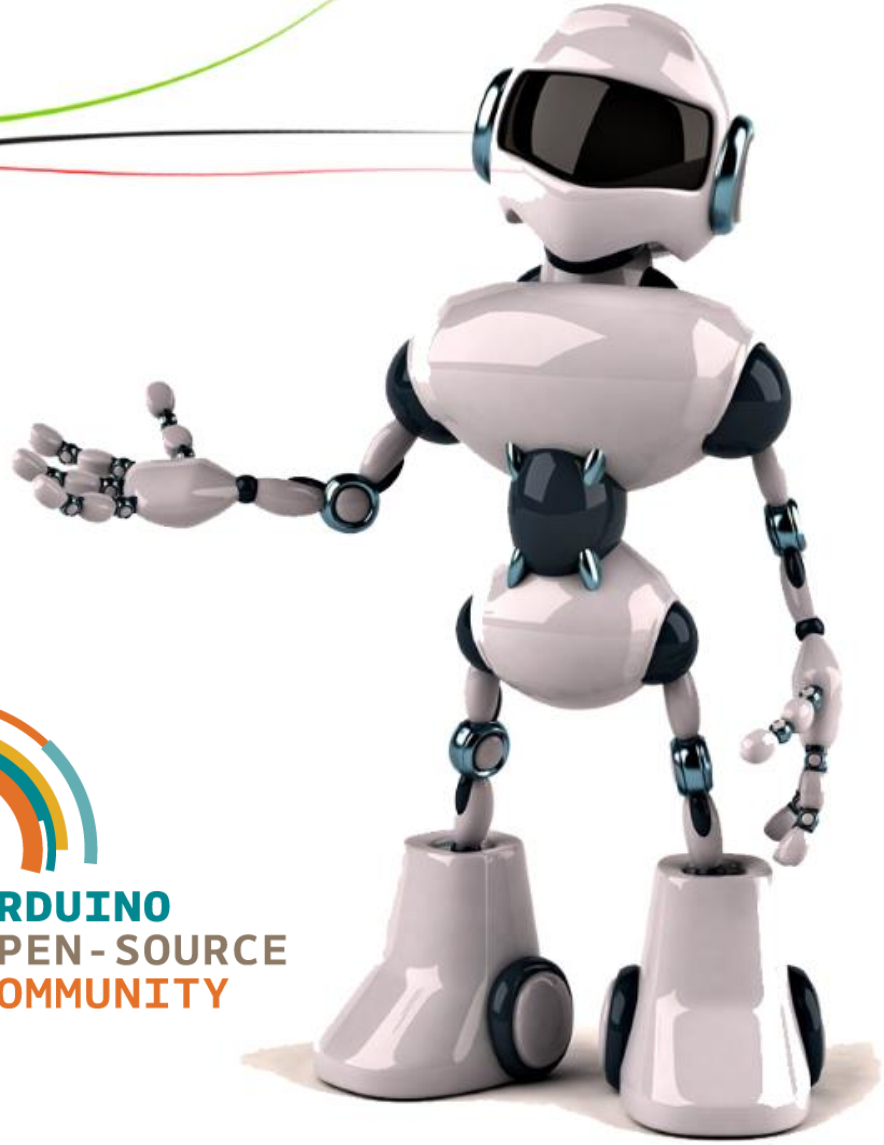
مکتب خونه

آکادمی آنلاین تخصص ها



ARDUINO
OPEN-SOURCE
COMMUNITY

دوره آموزشی آردوینو

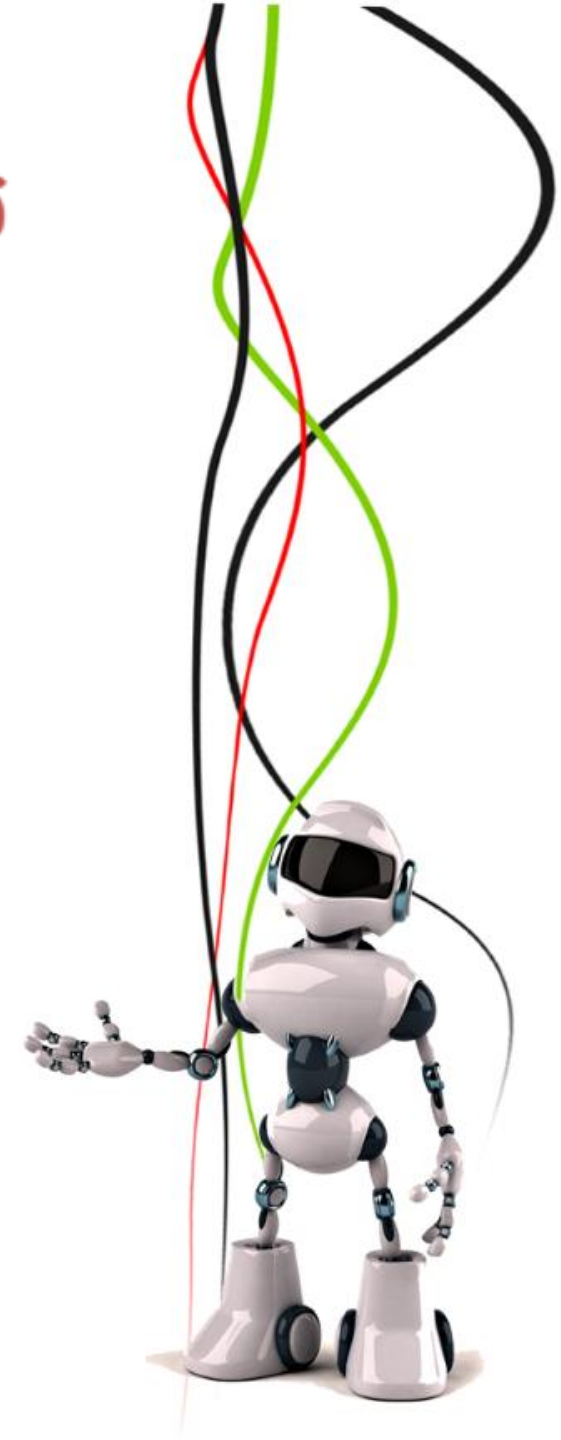
فصل دوم

بخش نهم : آرایه ها و توابع

مدرس : محمد پارسا کریمی

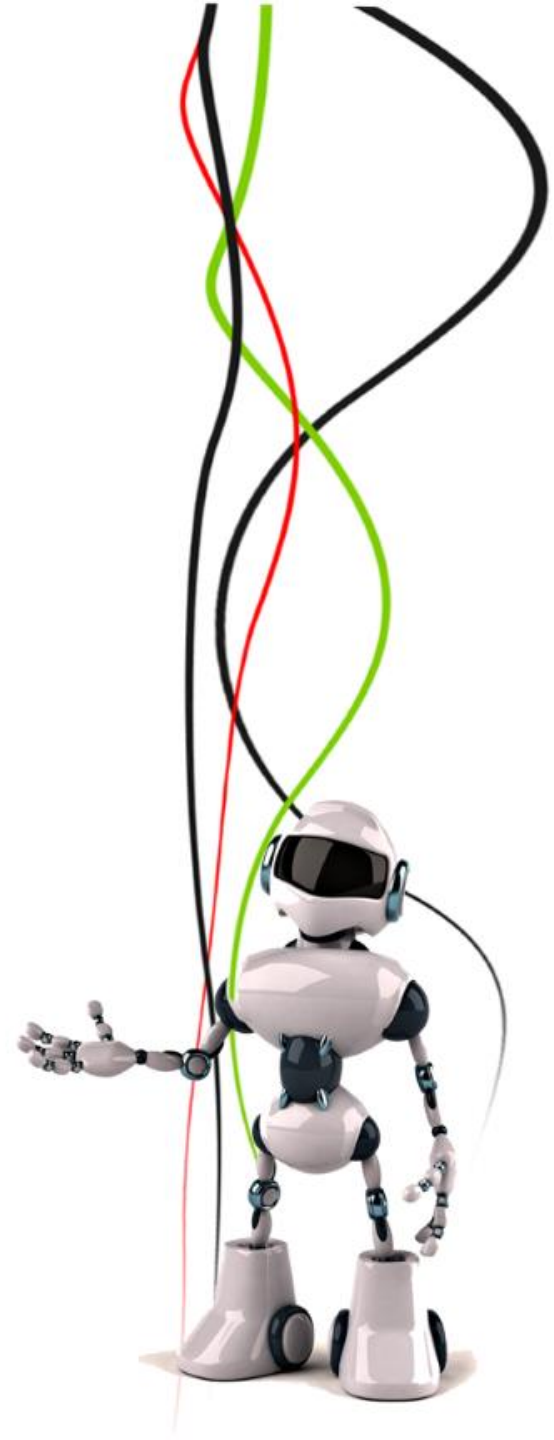
توابع

- برنامه‌هایی که تاکنون نوشته شده‌اند، فقط شامل یک تابع اصلی به نام main() بوده‌اند.
- برنامه‌های واقعی و تجاری بسیار بزرگتر از برنامه‌هایی هستند که تاکنون بررسی کردیم. برای این که برنامه‌های بزرگ قابل مدیریت باشند، برنامه‌نویسان این برنامه‌ها را به زیربرنامه‌هایی بخشبندی می‌کنند.
- این زیر برنامه‌ها تابع نامیده می‌شوند.
- در واقع تابع برنامه‌ای است که برای حل قسمتی از برنامه استفاده می‌شود.



توابع کتابخانه‌ای استاندارد

- تعدادی از توابع، که کاربرد بسیاری دارند و در اغلب برنامه‌ها مورد استفاده قرار می‌گیرند از قبل نوشته شده و به عنوان بخشی از محیط برنامه‌نویسی C++ ارائه می‌شوند.
- این توابع را توابع کتابخانه‌ای استاندارد C++ می‌نامند.
- در واقع کتابخانه استاندارد C++ یک مجموعه‌ی غنی از توابع را برای انجام محاسبات معمول ریاضی، دستکاری کاراکتر، ورودی/خروجی و بسیاری از اعمال مفید دیگر فراهم می‌آورد.(برای مثال تابع getline() که در فایل سرآیند <string> قرار داشت)
- در ادامه به بررسی برخی توابع کتابخانه‌ای ریاضی می‌پردازیم.



تابع جذر sqrt()

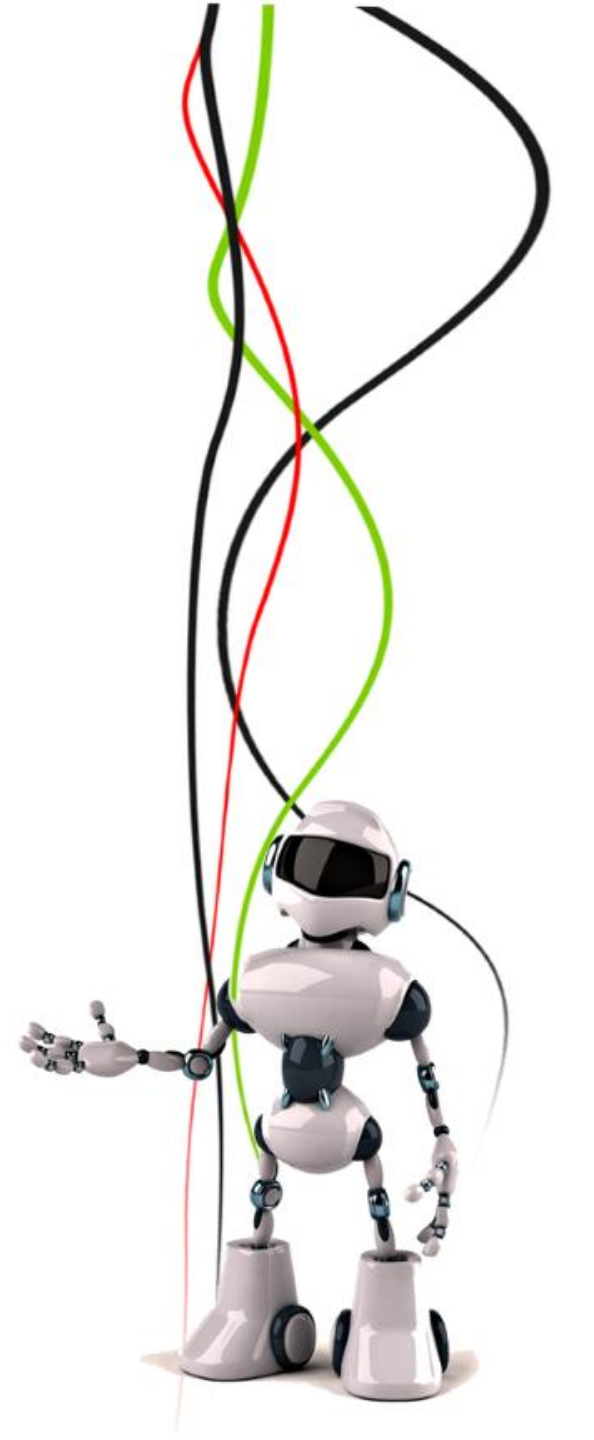
- فایل سرآیند <cmath> مجموعه توابعی که شما را قادر به انجام محاسبات ریاضی معمول می‌سازند، فراهم می‌آورد.
- تابع sqrt() برای محاسبه جذر یک عدد مثبت به کار می‌رود.
- نکته قابل توجه درمورد این تابع این است که ورودی باید از نوع double بوده خروجی نیز از نوع double می‌باشد.

A=sqrt(900.0) A | 30.0

- آرگومان‌های تابع می‌توانند ثابت، متغیر یا عبارات پیچیده‌تر باشند.
  - مثال: اگر c=13.0، d=3.0، f=4.0

```
cout<<sqrt(c+d*f);
```

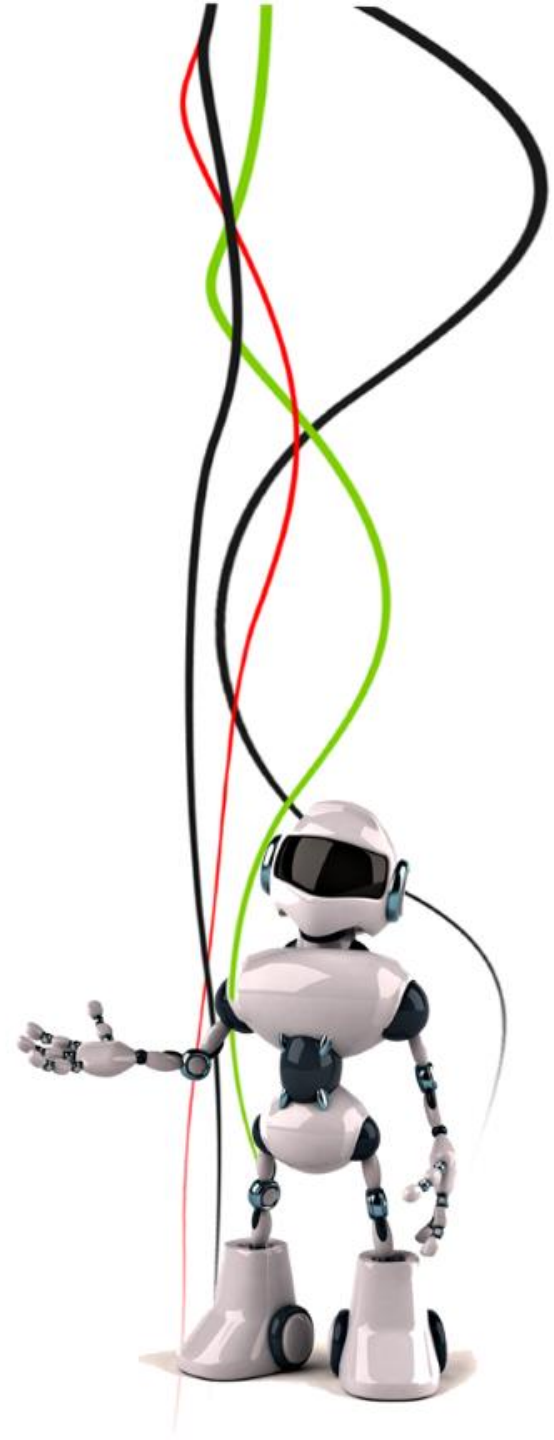
این دستور جذر 13.0+3.0*4.0=25.0 یعنی 5.0 را محاسبه می‌کند.



سایر توابع کتابخانه‌ای استاندارد

- بیشتر توابع معروف ریاضی که در ماشین‌حسابها هم وجود دارد در فایل سرآیند <cmath> تعریف شده‌اند.
- توجه کنید که تمامی این توابع مقدار double برمیگردانند و ورودی‌ها نیز باید از نوع double باشند.

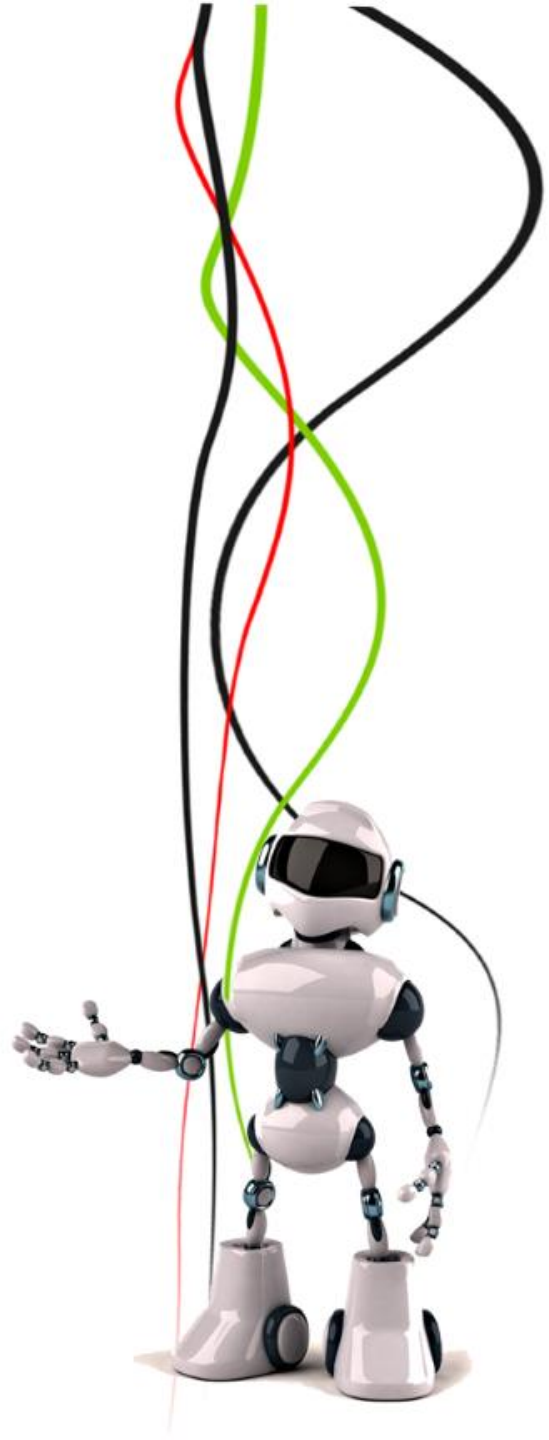
`ceil(x)`	مقدار سقف x (گرد شده)	`ceil(3.141593)` مقدار 4.0 را برمی‌گرداند
`cos(x)`	کسینوس x (به رادیان)	`cos(2)` مقدار 0.416147- را برمی‌گرداند
`exp(x)`	تابع نمایی x (در پایه e)	`exp(2)` مقدار 7.38906 را برمی‌گرداند
`fabs(x)`	قدر مطلق x	`fabs(-2)` مقدار 2.0 را برمی‌گرداند
`floor(x)`	مقدار کف x (گرد شده)	`floor(3.141593)` مقدار 3.0 را برمی‌گرداند
`log(x)`	لگاریتم طبیعی x (در پایه e)	`log(2)` مقدار 0.693147 را برمی‌گرداند
`log10(x)`	لگاریتم عمومی x (در پایه 10)	`log10(2)` مقدار 0.30103 را برمی‌گرداند
`pow(x,p)`	x به توان p	`pow(2,3)` مقدار 8.0 را برمی‌گرداند
`sin(x)`	سینوس x (به رادیان)	`sin(2)` مقدار 0.909297 را برمی‌گرداند
`sqrt(x)`	جذر x	`sqrt(2)` مقدار 1.41421 را برمی‌گرداند
`tan(x)`	تانژانت x (به رادیان)	`tan(2)` مقدار 2.18504- را برمی‌گرداند



نوشتن تابع

- گرچه توابع بسیار متنوعی در کتابخانه‌ی استاندارد C++ وجود دارد ولی این توابع برای بیشتر وظایف برنامه‌نویسی کافی نیستند. علاوه بر این برنامه‌نویسان دوست دارند خودشان بتوانند توابعی را بسازند و استفاده نمایند.
- برای نوشتن تابع باید اهداف تابع مشخص باشد. تابع چه وظیفه‌ای دارد، ورودی‌های تابع چیست و خروجی‌های تابع کدامند. با دانستن این موارد نوشتن تابع دشوار نیست.
- هر تابع دارای دو جنبه است، جنبه تعریف تابع و جنبه فراخوانی تابع.
- تعریف تابع: مجموعه دستوراتی که عملکرد تابع را مشخص می‌کنند.
- فراخوانی تابع: دستوری که تابع را صدا کرده و از آن استفاده می‌کند.

❖ نکته: فراخوانی تابع با نام آن صورت می‌گیرد و نامگذاری تابع از قوانین نامگذاری متغیرها استفاده می‌کند.



یک مثال ساده از توابع ساخت کاربر

این تابع، مکعب یک عدد صحیح ارسالی به آن را برمیگرداند.
بنابراین فراخوانی cube(3) مقدار ۹ را برمیگرداند.

عنوان تابع

```
int cube(int x)
```

بدنه تابع

```
{
// returns cube of x:
return x*x*x;
}
```

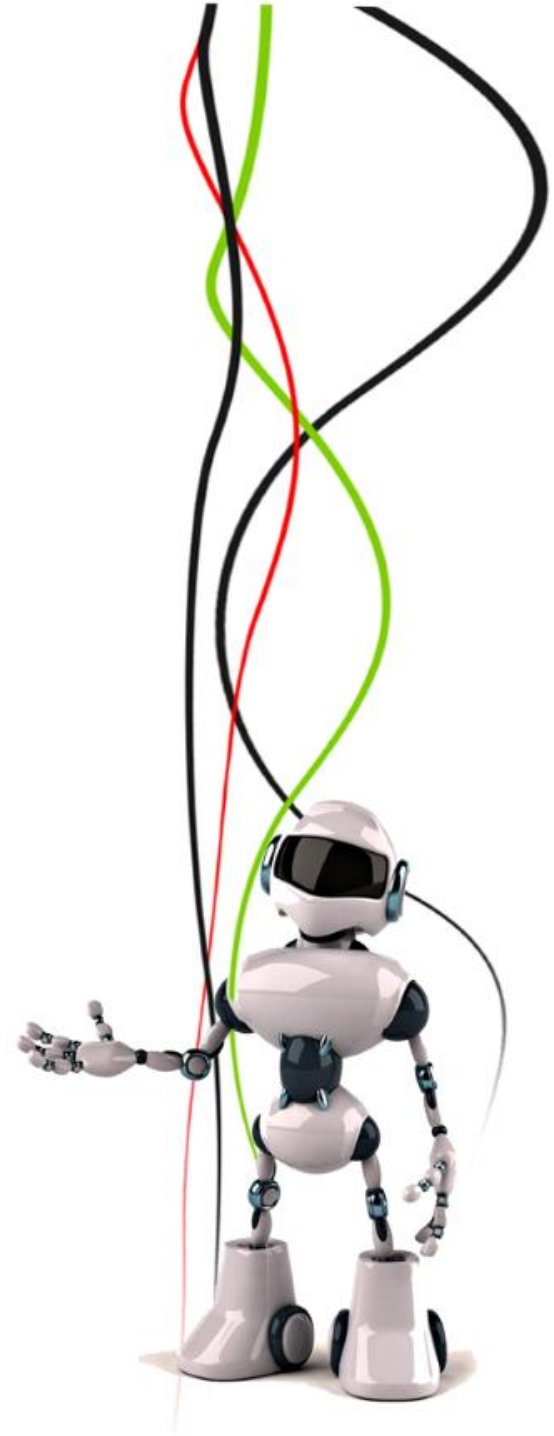
- یک تابع ساخت کاربر دو قسمت دارد: عنوان و بدنه.
- عنوان یک تابع به صورت زیر است:

نوع بازگشتی تابع نام تابع (فهرست پارامترها)

```
int          cube        (int  x)
```

این تابع یک ورودی از نوع int دارد.

خروجی این تابع مقداری از نوع int خواهد بود.

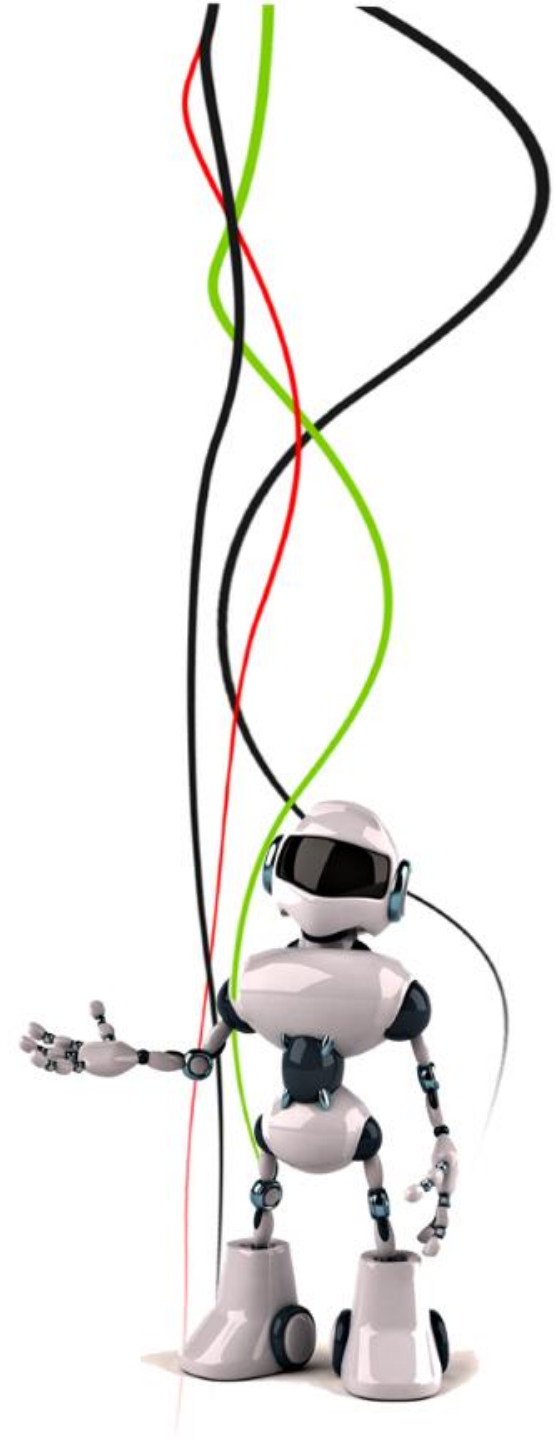


یک مثال ساده از توابع ساخت کاربر

```
int cube(int x)
{
// returns cube of x:
return x*x*x;
}
```

- بدنهٔ تابع، یک بلوک کد است که در ادامهٔ عنوان آن می‌آید.
- بدنه شامل دستوراتی است که باید انجام شود تا نتیجهٔ مورد نظر به دست آید.
- بدنه شامل دستور return است که پاسخ نهایی را به مکان فراخوانی تابع برمیگرداند.
- دستور return دو وظیفهٔ عمده دارد: اول این که اجرای تابع را خاتمه و دوم این که مقدار نهایی را به برنامهٔ فراخوان باز میگرداند

عبارت int main() که در همه برنامه‌ها استفاده کردیم یک تابع به نام «تابع اصلی» را تعریف میکند. مقدار بازگشتی این تابع از نوع int است و فهرست پارامترهای آن خالی است یعنی هیچ پارامتری نیاز ندارد.

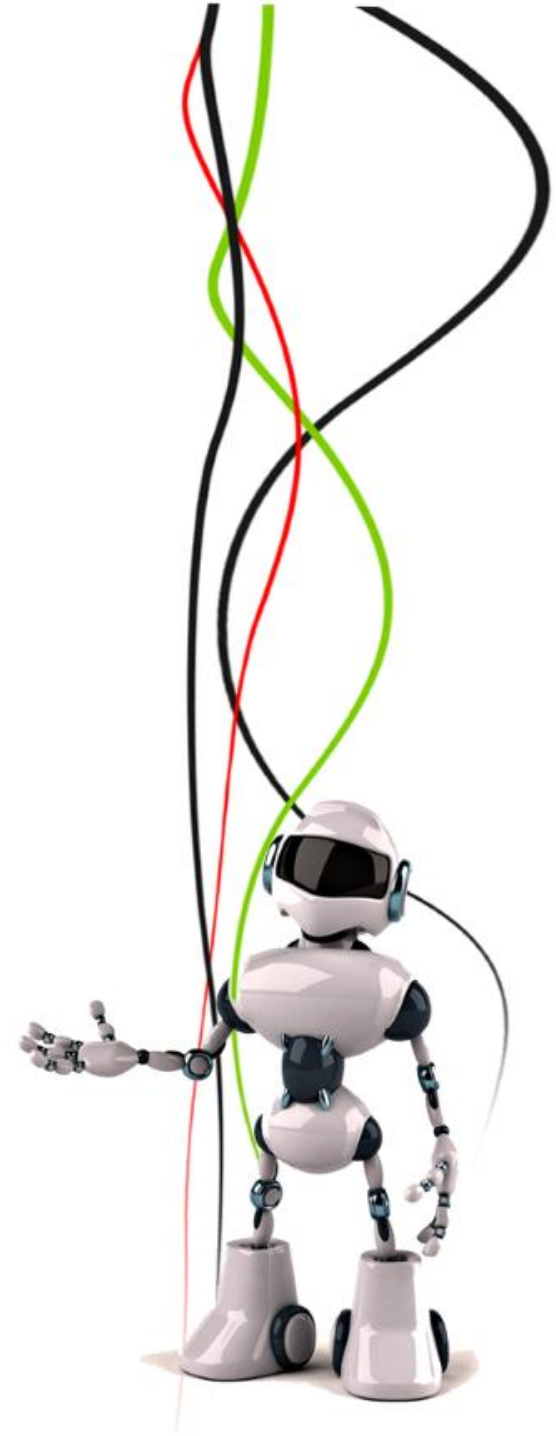


ساختار کلی توابع C++

- ساختار کلی توابع C++ در شکل زیر مشاهده میشود.

```
<نوع تابع>      نام تابع      (لیست پارامترها)
{
        <دستور ۱>
        <دستور ۲>
        .
        .
        .
        دستور n
}
```

- چنانچه تابع موردنظر هیچ مقداری را به عنوان خروجی برنگرداند، نوع تابع void تعریف میشود.

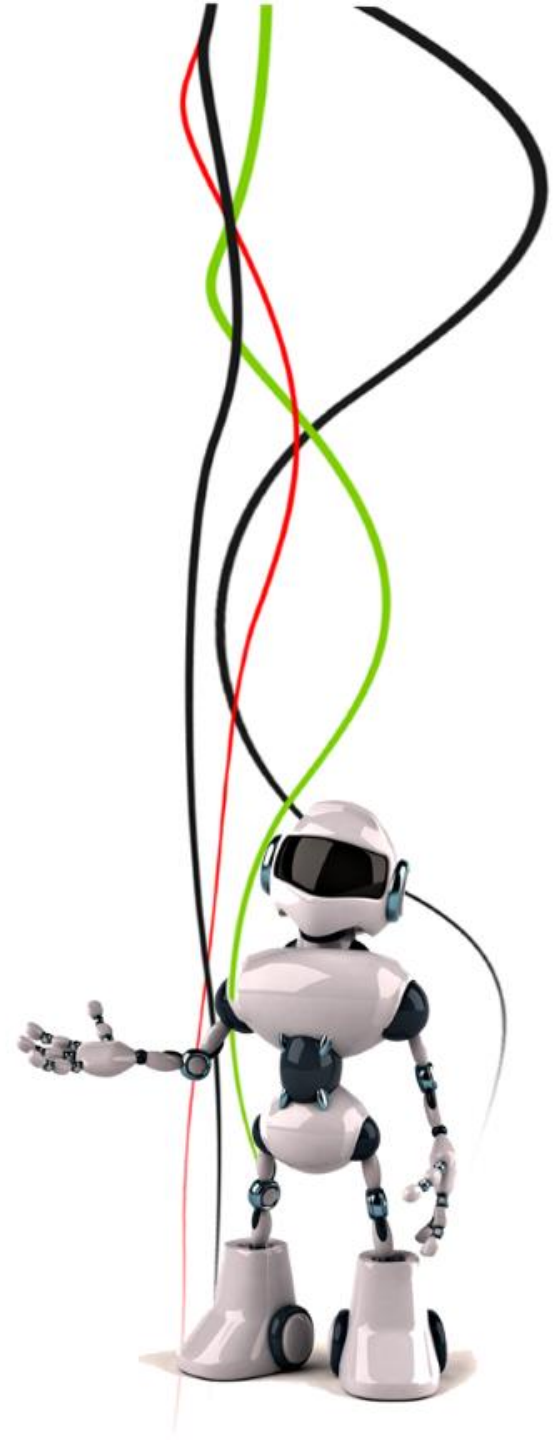


روش استفاده از یک تابع در برنامه اصلی

- هنگام استفاده از یک تابع میبایست الگوی آن تابع قبل از تابع main() اعلان شود.
- منظور از الگوی تابع همان عنوان تابع به همراه یک ; در انتهای آن است.

نوع بازگشتی تابع نام تابع (فهرست پارامترها);

- تابع مورد نظر به همراه دستوراتش میتواند پس از اتمام بدنه اصلی تابع تعریف شود.



روش استفاده از یک تابع در برنامه اصلی

```
#include <iostream>
int cube (int x);
int main()
{
   int answer;
   answer=cube(5);
   cout << answer ;
   return 0;
}// end of main

int cube(int x)
{
    return x*x*x;
}
```

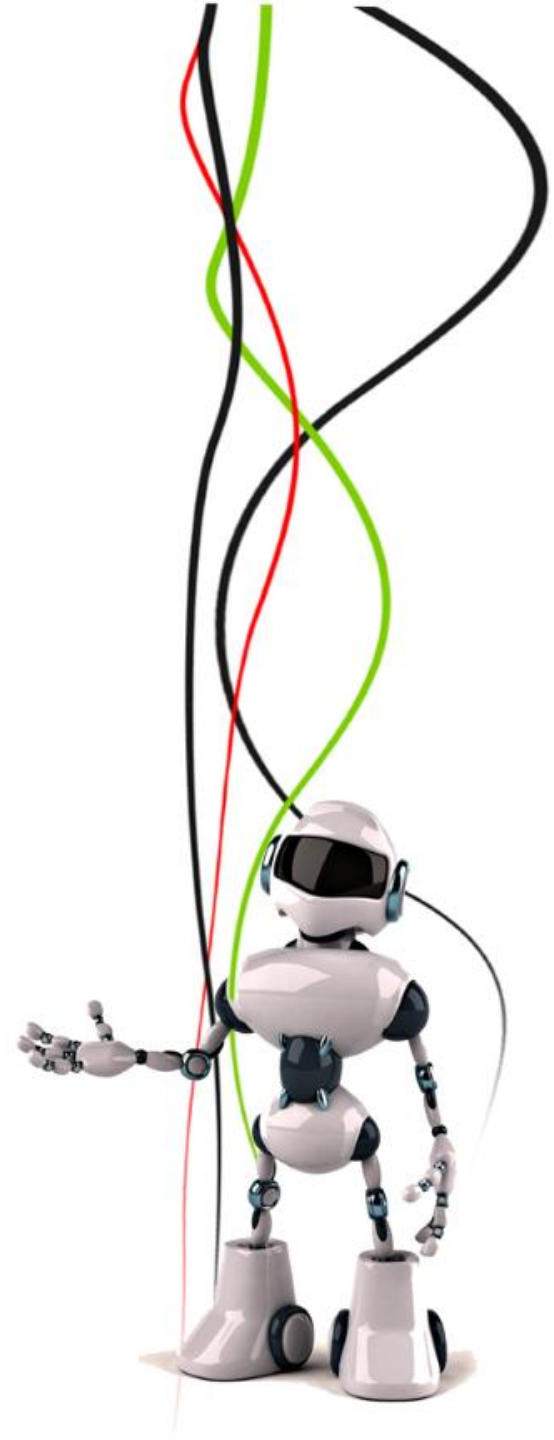
الگوی تابع cube

تابع اصلی main

فراخوانی تابع cube

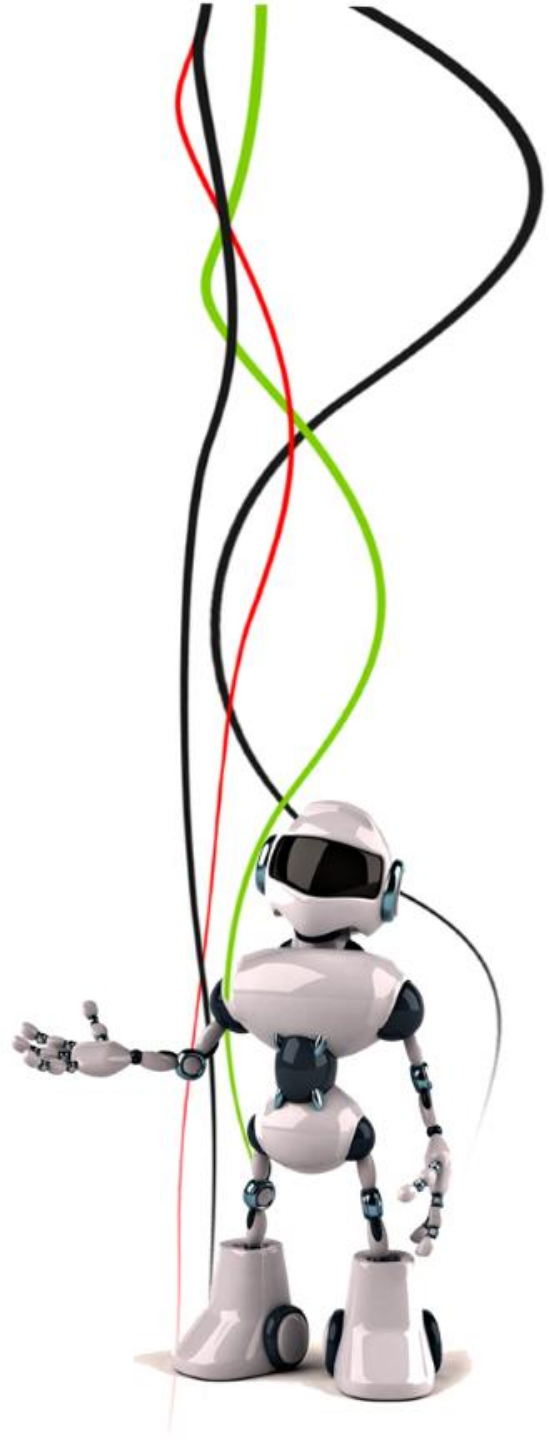
تابع cube. تعریف شده توسط برنامه نویس

نکته: چنانچه بدنه کامل تابع قبل از تابع main() نوشته شود دیگر نیازی به اضافه کردن الگوی تابع نمی‌باشد. البته این روش زیاد معمول نیست.



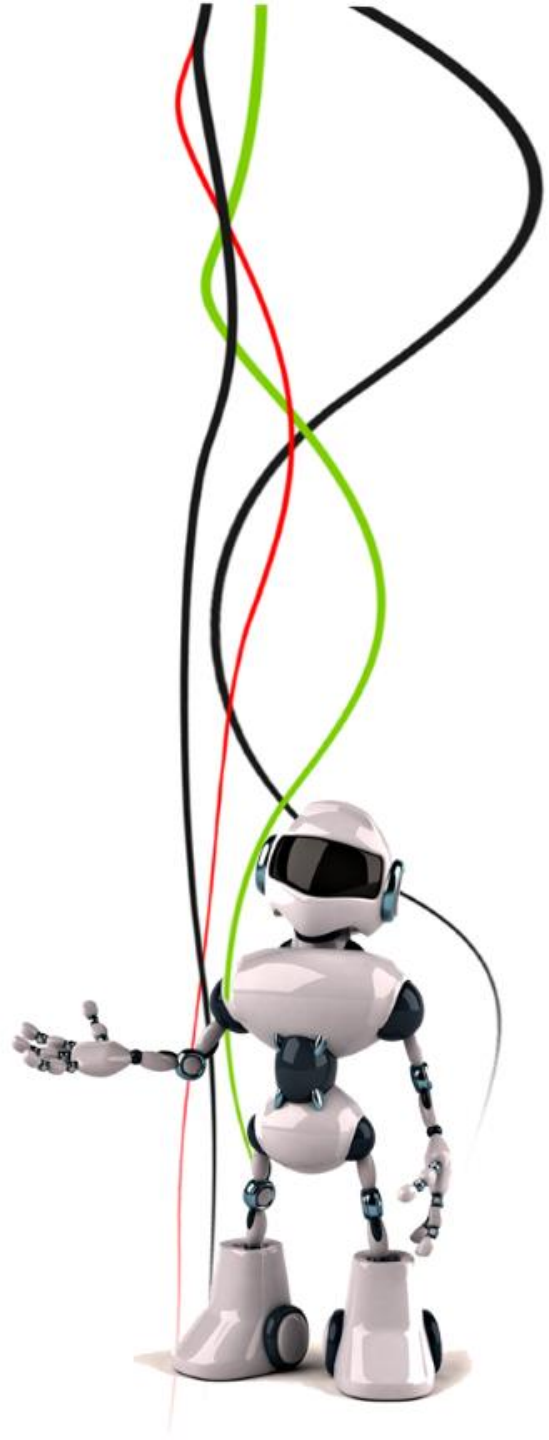
متغیرهای محلی و عمومی

- متغیر محلی (local variable) متغیری است که در داخل یک بلوک اعلان می‌شود و تنها در داخل همان بلوک قابل دسترسی می‌باشد.
- متغیرهایی که در داخل بلوک یک تابع تعریف میشوند نیز محلی بوده و تنها داخل همان تابع قابل استفاده می‌باشند.
- بنابراین هر متغیری که در داخل تابع ()main تعریف می‌شود تنها در همین تابع قابل استفاده می‌باشد و در داخل توابع دیگر شناخته شده نیست مگر اینکه از طریق آرگومانها به توابع دیگر ارسال شوند.
- حال چنانچه تعریف یک متغیر در خارج از توابع و در بالای تابع ()main تعریف شوند در تمام توابع موجود در برنامه قابل استفاده‌اند و متغیر عمومی(global variable) نامیده میشوند.



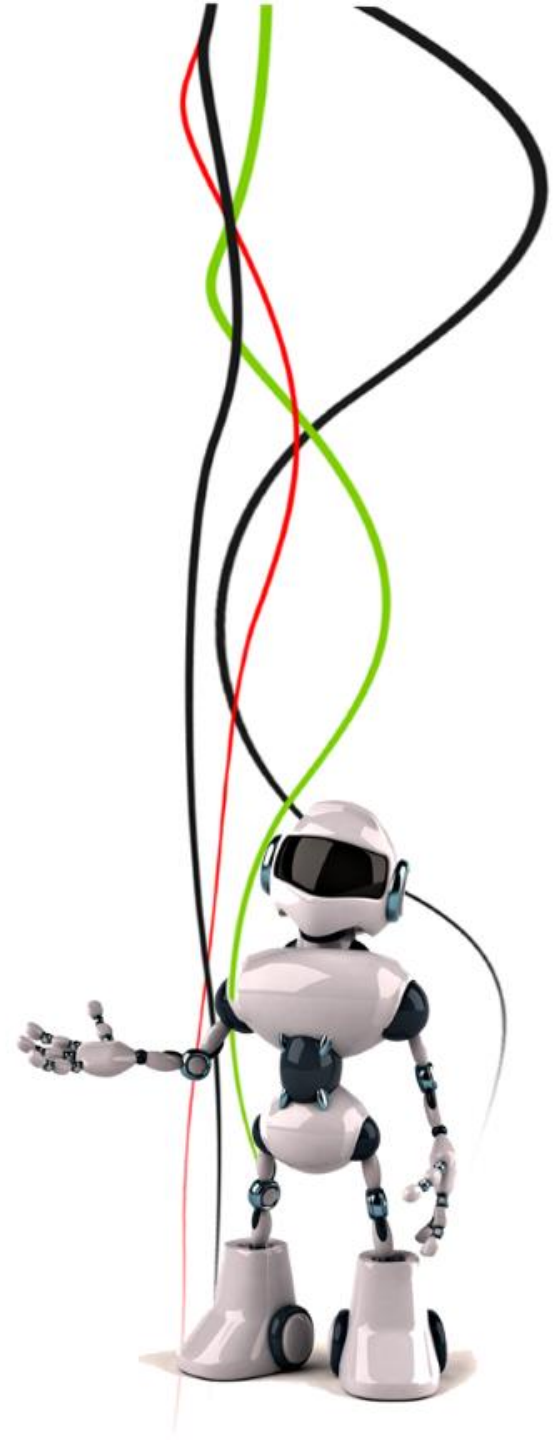
متغیرهای محلی و عمومی(نکات)

- با توجه به اینکه از طریق پارامترها می‌توان بین پارامترها ارتباط برقرار کرد معمولاً نیازی به استفاده از متغیرهای عمومی نمی‌باشد.
- متغیرهای عمومی درک و نگهداری برنامه را دشوار می‌کنند، بنابراین توصیه می‌شود حدلامکان از متغیرهای عمومی استفاده نشود، مگر اینکه در یک برنامه اکثر توابع بخواهند از یک متغیر استفاده کنند.
- استفاده از ثابت‌های عمومی به دلیل عدم امکان تغییر آنها توسط توابع، اثرات جانبی چندانی ندارد و لذا استفاده از ثوابت عمومی از متغیرهای عمومی مرسوم تر است.



آرایه‌ها

- یک متغیر بخشی از حافظه است که یک نام دارد و میتوان مقداری را در آن ذخیره کرد.
- در برنامه‌های کوچک ممکن است بتوانیم کل پردازش را با استفاده از متغیرها عملی کنیم ولی در برنامه‌هایی که داده‌های فراوانی را پردازش میکنند استفاده از متغیرهای معمولی کار عاقلانه‌ای نیست زیرا در بسیاری از این برنامه‌ها «پردازش دسته‌ای» صورت میگیرد
- برای این منظور از آرایه‌ها استفاده می‌شود.
- آرایه را میتوان متغیری تصور کرد که یک نام دارد ولی چندین مقدار را به طور همزمان نگهداری مینماید

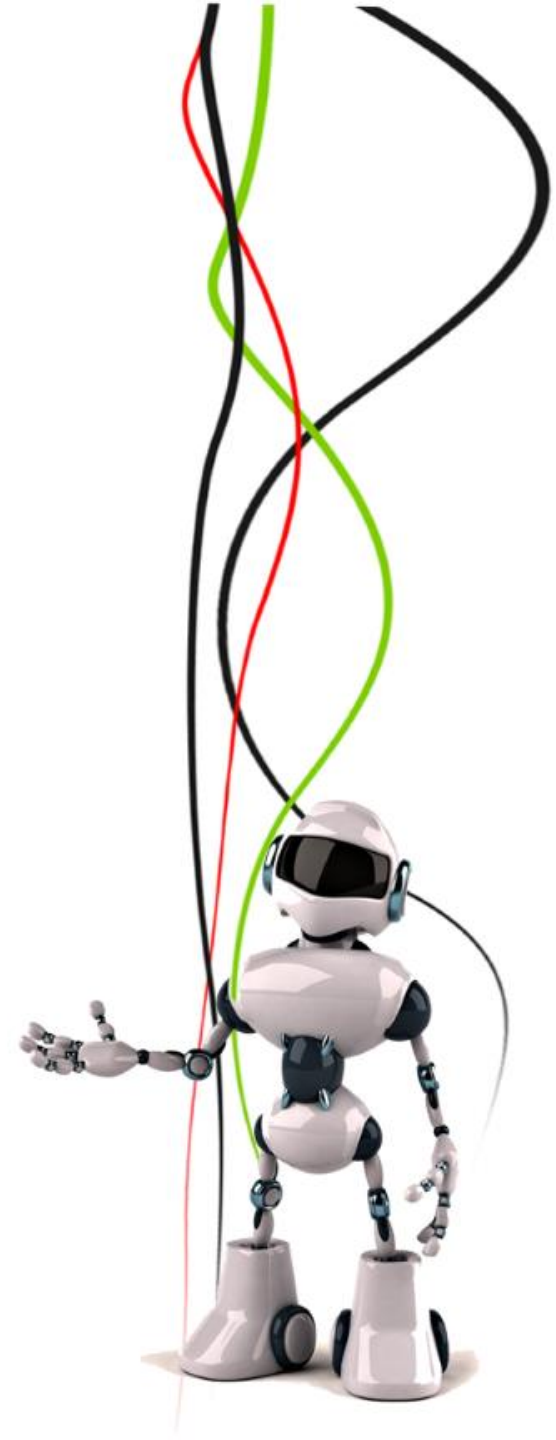


آرایه‌ها

- یک آرایه، یک زنجیره از متغیرهایی است که همه از یک نوع هستند.
- هر آرایه دارای نامی است و مانند متغیرهای معمولی نامگذاری میشود.
- هر عضو آرایه با یک عدد مشخص میشود که به آن «اندیس» می‌گویند. این عدد شماره عناصر از ابتدای آرایه مشخص میکند.
- اعلان یک آرایه به صورت زیر میباشد:

```
نوع     نام آرایه     [اندازه آرایه];
int        x             [5];
```

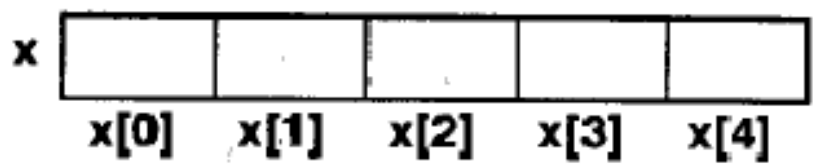
- این تعریف آرایه ای از نوع int و دارای ۵ خانه را تعریف میکند.

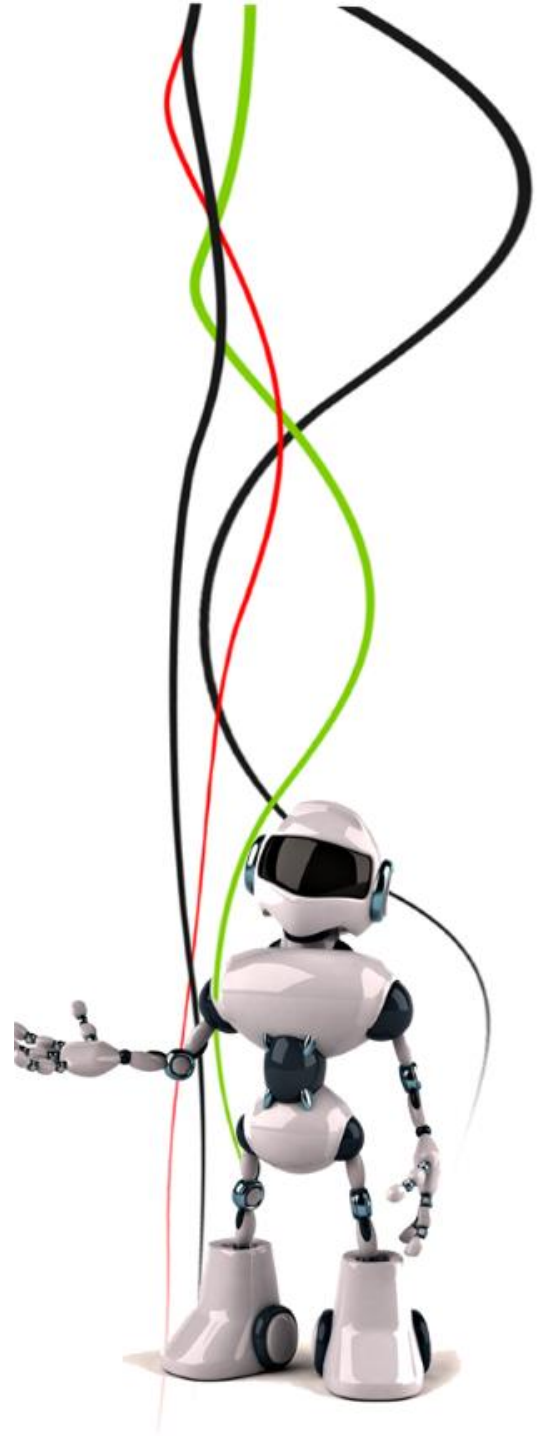


آرایه‌های یک بعدی

- نوع آرایه یکی از انواع قابل قبول در c++ است.
- نام آرایه برای دسترسی به عناصر آرایه مورد استفاده قرار میگیرد.
- طول آرایه با یک عدد صحیح مثبت (یا یک عبارت صحیح) مشخص میشود.
- اندیس خانه‌های آرایه از صفر شروع میشوند. به این ترتیب عناصر آرایه x به صورت زیر ارزیابی میشوند.
- توجه داشته باشید که عناصر آرایه در محلهای متوالی حافظه ذخیره میشوند.

```
نوع    نام آرایه    [اندازه آرایه];
int       x           [5];
```





مثال ۱: اعلان یک آرایه و استفاده از حلقه برای مقداردهی

- این برنامه با یک حلقه در داخل خانه‌های مختلف آرایه مقادیر قرار میدهد.

```
int main()
{
    int n[10];
  for(int i=0; i<10; i++){
    n[i]=i+1;
    cout << i<<setw(5)<<n[i]<<endl;}
    return 0;
}// end of main
```