1

درس برنامه نویسی

(آشنایی با زبان برنامه نویسی C++)

(رشته مهندسی برق – کنترل)

مدرس:
جواد عبدی

تعداد واحد درسی : ۲واحد
مهر ۹۲

2

دلیل انتخاب درس؟

معرفی درس:

- هدف ارایه مطالب به صورت کاربردی
- برنامه نویسی به عنوان ابزار یک مهندس یا محقق
- نامحدود بودن علم و امکان پذیر نبودن داشتن همه آن
- فرض: آشنايي دانشجوها با يك زبان برنامه نويسي

3

توصيه هاي مهم:

- در اين درس تنها روش ياد گيري تمرين است. لذا تمرين هاي زيادي ارايه خواهد شد.
- انتظار دريافت پاسخ در مورد هر سوالي را نداشته باشيد.
- آموزش در كلاس. سعي كنيد كليه مطالب را در كلاس متوجه شويد و پس از كلاس پياده كنيد (در مورد مثال ها).

4

نمره بندی:

۱- تمرین **۳ نمره**

۲- میان ترم یا پروژه برنامه نویسی **۵ نمره**

۳- پایان ترم **۱۲ نمره**

جهت ارسال تمرین ها : فقط و فقط به آدرسهای ایمیل

kai_ins@yahoo.com; hamidreza.chegini@gmail.com

زمان مجاز جهت ارسال:

روز قبل از تشکیل کلاس بعدی (یک هفته) – غیر قابل تمدید

حتماً شماره تمرین در Subject قید شود.

تمرین ها بصورت انفرادی هستند.

نام گذاری تمرین ها:

Yourname_HW3.doc **Yourname_HW3.cpp**

مثال: kiani_HW1

5

فهرست مطالب فصل اول

6

شروع درس:

کامپیوتر و سیستم بر عهده دانشجو
دانش ابتدایی از سیستم های کامپیوتری و سخت افزار آن به عهده دانشجو است

7

مقدمه:

زبان C یک زبان همه منظوره است. دستورالعمل‌های این زبان بسیار شبیه عبارات جبری و نحو آن شبیه جملات انگلیسی می باشد. این امر سبب می‌شود که C یک زبان سطح بالا باشد که برنامه‌نویسی در آن آسان است.

چرا زبان C؟

- **مزایای زبان C :**
  - **نسبت به سایر زبان های برنامه نویسی بسیار سریع و فشرده**
  - **بی نهایت قابل انعطاف**
  - **مقدمه بسیار خوبی برای یادگیری زبان هایی مانند C# , C++, JAVA,JAVA SCRIPT و غیره**

8

مزایای برنامه نویسی و استفاده از کامپیوتر:

انجام کارهای تکراری بصورت دقیق با سرعت فراوان

مثال:

محاسبه حقوق کارمندان یک شرکت

9

چرا C++

C++ که از نسل C است، تمام ویژگی‌های C را به ارث برده است. اما برتری فنی دیگری هم دارد: C++ اکنون «شی‌گرا» است. می‌توان با استفاده از این خاصیت، برنامه‌های شی‌گرا تولید نمود. برنامه‌های شی‌گرا منظم و ساخت‌یافته‌اند، قابل روزآمد کردن‌اند، به سهولت تغییر و بهبود می‌یابند و قابلیت اطمینان و پایداری بیشتری دارند.

10

فصل اول

مقدمات C++

تاریخچه مخترع زبان برنامه نویسی C

دنیس ریچی مخترع زبان برنامه‌نویسی C و یکی از اعضای گروه توسعه‌دهنده سیستم‌عامل یونیکس در آزمایشگاه بل در سن هفتاد سالگی درگذشت.
بخش قابل‌توجهی از فناوری‌های مدرن مرهون کارهای او و دیگر برنامه‌نویسانی است که در روزهای اولیه انقلاب کامپیوتر، یونیکس و C را پدید آوردند.

نفوذ یونیکس در دنیای کامپیوتر نفوذی چندجانبه بود و باعث ایجاد چندین اصل در مهندسی نرم‌افزار شد که تا امروز نیز اعتبار دارند. یونیکس سیستم‌عامل منتخب اینترنت و سرآغاز جنبش اپن‌سورس بود و توانست خود را با گونه‌های مختلف سخت‌افزار هماهنگ کند.
ریچی در سال ۱۹۶۷ به آزمایشگاه بل که در آن زمان یکی از بزرگ‌ترین ارائه‌دهندگان خدمات ارتباطی در آمریکا بود پیوست و بیشتر وقت خود را در آنجا گذراند و در سال ۲۰۰۷ بازنشسته شد.

تاریخچه مختصر C++

این زبان در اوائل دهه ۱۹۸۰ توسط **بیارنه استراستروپ** (Bjarne stroustrup) در آزمایشگاه بل طراحی شده. این زبان عملاً توسعه یافته زبان برنامه نویسی C می باشد که امکان نوشتن برنامه‌های ساخت یافته شئ گرا را می‌دهد.

زبان برنامه‌نویسی C++ (سی پلاس پلاس) یک زبان برنامه‌نویسی کامپیوتری عمومی با قابلیت‌های سطح بالا و سطح پایین می‌باشد. این زبان دارای قابلیت‌های انواع داده ایستا، نوشتار آزاد، چندمدلی، معمولاً زبان ترجمه شده با پشتیبانی از برنامه‌نویسی ساخت‌یافته، برنامه‌نویسی شی‌گرا، برنامه نویسی جنریک است. زبان C++یک زبان سطح میانی در نظر گرفته می‌شود. این زبان دارای قابلیت زبان‌های سطح بالا و پایین بصورت همزمان است. این زبان در سال ۱۹۷۹ در آزمایشگاه های بل (Bell Labs) و بر مبنای زبان سی ساخته شد و آن را " Cبا کلاس" نامگذاری نمودند. در سال ۱۹۸۳ به C++تغییر نام داد. توسعه با اضافه نمودن کلاس‌ها و ویژگی‌های دیگری مانند توابع مجازی، سربارگزاری عملگرها، وراثت چندگانه، قالب توابع، و پردازش استثنا انجام شد. این زبان برنامه‌نویسی در سال ۱۹۹۸ تحت نام ISO/IEC 14882:1998استاندارد شد.

تاریخچه مختصر C++ (ادامه)

کامپایلرها و IDE های گوناگونی برای زبان C++وجود دارند از بین معروفترین آن ها می توان موارد زیر اشاره نمود:

Turbo C: یکی از کامپایلرهای قدیمی زبان برنامه نویسی C است اما با وجود اینکه مدت ها از تاریخ انتشار آن میگذرد همچنان یکی از محبوب ترین کامپایلرها به شمار می‌رود. این نرم افزار دارای ظاهر جالبی نبوده و همواره محیطی مشابه محیط داس را برای کاربران تداعی می کند چرا که از ابتدا این کامپایلر برای سیستم عامل داس نوشته شده است.

Turbo C++: یکی از زبان های برنامه نویسی قدرتمند است. این زبان در تمامی سیستم عامل‌ها اجرا می‌کند.

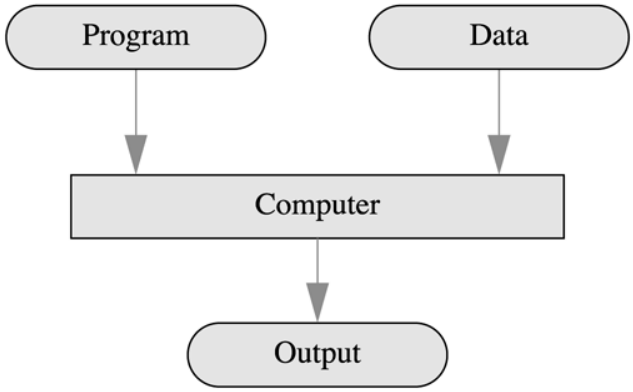
Borland C++: نام یکی از کامل‌ترین و مشهورترین نسخه های زبان های برنامه نویسی تحت ویندوز برای توسعه نرم افزارهای کاربردی و بانک‌های اطلاعاتی می باشد. این نرم افزار برنامه نویسی قابلیت پشتیبانی از سیستم عامل های جدید ویندوز تا ویندوز ۷ را دارا می باشد و در آن از تکنولوژی Client-Server استفاده شده است و امکان برنامه نویسی بروی Linux نیز در این زبان برنامه نویسی وجود دارد.

تاریخچه مختصر C++ (ادامه)

Microsoft C++/C: زبان C++/C یکی از قدرتمندترین زبان های برنامه نویسی می باشد. منظور از C++/C یعنی هم زبان C و هم زبان C++ است C++ .خود فرزند C است و همه قابلیت های C به علاوه برخی قابلیت های جدید مثل شی ء گرایی را دارا می باشد، در نتیجه، کامپایلر هایC++ ، کد نوشته شده به زبان C را نیز می توانند کامپایل کنند. زبان C++/C وابسته به یک سیستم عامل نیست یعنی شما بعد از نوشتن برنامه خود به زبانC++/C، اگر کد استانداری نوشته باشید می توانید با توجه به سیستم عامل (ویندوز، گنو/لینوکس، مک و...)، کدتان را کامپایل کنید. می توان کد C++/Cرا در هر محیطی (مثلاً NotePad در ویندوز و یا gEdit در گنو/لینوکس) نوشته و بعد آن را بوسیله یک کامپایلر کامپایل کنیم، ولی برای راحتی کار ما می توانیم از یک IDE مناسب، نیز بهره ببریم.

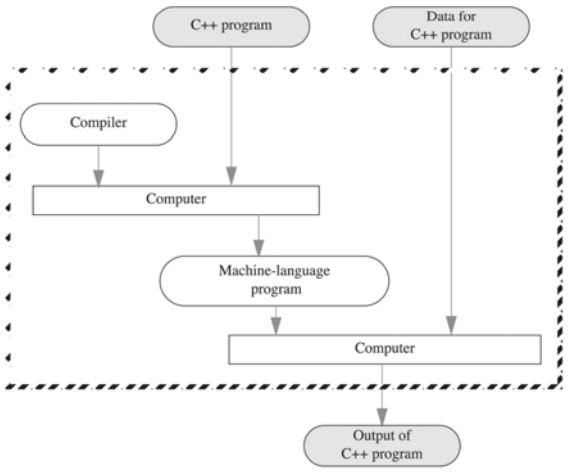
نگاهی به نحوهٔ اجرای برنامه در زبان C++

Simple View of Running a Program

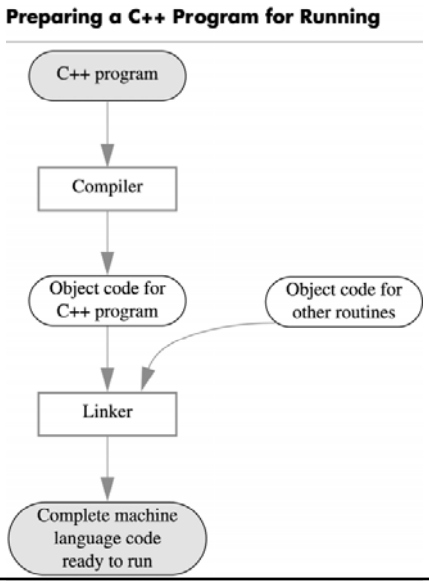


نگاهی به نحوهٔ اجرای برنامه در زبان C++ - ادامه

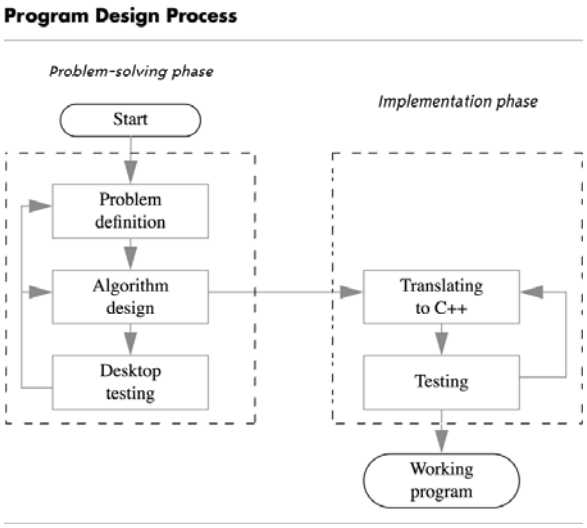
Compiling and Running a C++ Program (Basic Outline)



نگاهی به نحوهٔ اجرای برنامه در زبان C++ - ادامه



نگاهی به نحوهٔ اجرای برنامه در زبان C++ - ادامه



قانون نامگذاری شناسه‌ها

حروف کوچک و بزرگ در نامگذاری شناسه‌ها متفاوت می‌باشند.

بنابراین Xy ، XY ، xY ، xy چهار شناسه متفاوت از نظر C++ می‌باشد.

Case Sensetive

قانون نامگذاری شناسه‌ها

در نامگذاری شناسه‌ها از حروف الفباء، ارقام و زیر خط (underscore) استفاده می‌شود و حداکثر طول شناسه ۳۱ می‌باشد و شناسه بایستی با یک رقم شروع نگردد.

قانون نامگذاری شناسه‌ها

برای نامگذاری شناسه‌ها از کلمات کلیدی نبایستی استفاده نمود. در زیر بعضی از کلمات کلیدی داده شده است.

And	Sizeof	then	xor	Template
Float	False	Friend	While	continue
extern	Private	Switch	Default	Const
delete	typedef	if	this	Virtual

لیست کامل کلمات کلیدی

متغیرها

متغیر، مکانی در حافظه اصلی کامپیوتر می‌باشد که در آنجا یک مقدار را می‌توان ذخیره و در برنامه از آن استفاده نمود. قانون نامگذاری متغیرها همان قانون نامگذاری شناسه‌ها

در اسلاید بعد به انواع داده ها اشاره می شود.

انواع داده ها

نوع داده	مقادیر	حافظه لازم
int	32767 تا 32768–	۲ بایت
unsigned int	65535 تا 0	۲بایت
long int	2147483647 تا 2147483648–	۴ بایت
unsigned long int	4294967295 تا 0	۴ بایت
char	یک کارکتر	۱بایت
signed char	127 تا 128–	۱ بایت
unsigned char	256 تا 0	۱ بایت
float	1.2e-38 تا 3.4e38	۴ بایت
double	2.2e-308 تا 1.8e308	۸ بایت

اعلان یا فراخوانی متغیرها

قبل از آنکه در برنامه به متغیرها مقداری تخصیص داده شود و از آنهـا استفاده گردد بایستی آنها را در برنامه اعلان نمود.

چند مثال از اعلان متغیر ها :

✓ برای اعلان متغیرx از نوع int :

```
int   x;
```

✓ برای اعلان متغیرهای p و q را از نوع float که هر کدام چهار بایت از حافظه را اشغال می‌کنند :

```
float p , q ;
```

✓ برای اعلان متغیر next از نوع کاراکتر که می‌توان یکی از ۲۵۶ کاراکتر را به آن تخصیص داد و یک بایت را اشغال می‌کند.

```
char next ;
```

25

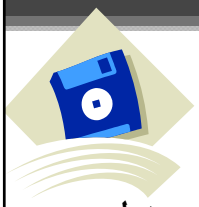
تخصیص مقادیر به متغیرها

با استفاده از عملگر = می‌توان به متغیرها مقدار اولیه تخصیص نمود.

مثال :

- در دستورالعمل `int x=26;` x را از نوع int با مقدار اولیه 26 اعلان نموده است.
- در دستورالعمل `long a=67000 , b=260;` متغیرهای b و a را از نوع long int تعریف نموده با مقادیر بترتیب 260 و 67000.

26



داده‌های از نوع کاراکتر

برای نمایش داده‌های از نوع char در حافظه کامپیوتر از جدول ASCII استفاده می‌شود. جدول اسکی به هر یک از ۲۵۶ کاراکتر یک عدد منحصر بفرد بین ۰ تا ۲۵۵ تخصیص می‌دهد.

کامپایلر C++ بعضی از کاراکترهای مخصوص که در برنامه می‌توان از آنها برای فرمت بندی استفاده کرد را تشخیص می‌دهد. تعدادی از این کاراکترهای مخصوص به همراه کاربرد آنها در اسلاید بعد آورده شده است .

کاراکترهای مخصوص

\n	Newline
\t	Tab
\b	Backspace
\a	Beep sound
\”	Double quote
\’	Single quote
\0	Null character
\?	Question mark
\\	Back slash

بعنوان مثال از کاراکتر \a می‌توان برای ایجاد صدای زنگ (beep) استفاده نمود.

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


رشته‌ها

رشته یا string عبارتست از دنباله‌ای از کاراکترها که بین " " قرار داده می‌شود. در حافظه کامپیوتر انتهای رشته‌ها بوسیله 0\ ختم می‌گردد.

مثال ۱ :

"BOOK STORE" یک رشته ده کاراکتری می‌باشد که با توجه به کاراکتر 0\ که به انتهای آن در حافظه اضافه می شود جمعاً یازده بایت را اشغال می‌کند.

مثال ۲:

دقت نمایید که "w" یک رشته می‌باشد که دو بایت از حافظه را اشغال می‌کند در حالیکه 'w' یک کاراکتر می‌باشد که یک بایت از حافظه را اشغال می‌نماید.

29

نمایش مقادیر داده‌ها

برای نمایش داده‌ها بر روی صـفحه مـانتور از **cout** کـه بـدنبال آن عملگـر درج یعنی >>قید شده باشد استفاده می‌گردد. بایستی توجـه داشـت کـه دوکـاراکتر < پشت سر هم توسط C++ بصورت یک کاراکتر تلقی می‌گردد.

مثال :

✓ برای نمایش پیغام good morning بر روی صفحه نمایش :

```
cout << "good morning";
```

✓ برای نمایش مقدار متغیر X بر روی صفحه نمایش :

```
cout << x ;
```

30

cout << variable-name;
Meaning: print the value of variable <variable-name> to the user
cout << "any message ";
Meaning: print the message within quotes to the user
cout << endl;
Meaning: print a new line

Example:

```
cout << a;
cout << b << c;
cout << "This is my character: " << my-character << " he he he"
     << endl;
```

31

دریافت مقادیر متغیرها

به منظور دریافت مقادیر برای متغیرها در ضمن اجرای برنامه از صفحه کلید، از **cin** که بدنبال آن عملگر استخراج یعنی >> قید شده باشد می‌توان استفاده نمود.

مثال :

```
int x;
cout << "Enter a number:" ;
cin >> x;
```

32

تمرین ۱ مهم HW1

خروجی برنامه های زیر را پس از اجرا یادداشت نمایید.

```
#include <iostream.h>
int main()
{
    cout << "Salam!";
    return 0;
}
```

```
#include <iostream.h>
int main()
{
    cout << "Salam!";
    cout << "Salam!";
    cout << "Salam!";
    return 0;
}
```

33

تمرین ۱ مهم HW1

خروجی برنامه های زیر را پس از اجرا یادداشت نمایید.

```
#include <iostream.h>
int main()
{
	int a;
	a = 5;
	cout << "a=" << a << "\n";
	return 0;
}
```

```
#include <iostream.h>
int main()
{
	int a;
	int b, c;
	b = 2;
	c = 3;a = (b + c + 11) / 3;
	cout << "hasel=" << a << "\n";
	return 0;
}
```

34

عملگر انتساب

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

35

مثال :

```
x=a+b;
 x=35 ;
 x=y=z=26 ;
```

از عملگرهای انتساب چندگانه نیز می‌توان استفاده نمود. که مقدار سه متغیر z و y و x برابر با ۲۶ می شود.

عملگرهای محاسباتی

در C++ پنج عملگر محاسباتی وجود دارد که عبارتند از :

+	**جمع**
–	**تفریق**
*	**ضرب**
/	**تقسیم**
٪	**باقیمانده**

این عملگرها دو تائی می‌باشند زیرا روی دو عملوند عمل می‌نمایند. از طرف دیگر عملگرهای + و – رامی‌توان بعنوان عملگرهای یکتائی نیز در نظر گرفت.

مثال ۱ :

در حالتی که هر دو عملوند عملگرهای ٪ ، / ، * ، + ، – از نوع صحیح باشد نتیجه عمل از نوع صحیح می‌باشد.

عبارت	نتیجه
5 + 2	7
5 * 2	10
5 – 2	3
5 % 2	1
5 / 2	2

مثال ۲:

در صورتیکه حداقل یکی از عملوندهای عملگرهای / ، * ، – ، + از نوع اعشاری باشد نتیجه عمل از نوع اعشاری می‌باشد. در اعداد اعشاری تقسیم برگشت داده نمی شود.

عبارت	نتیجه
5. 0 +2	7. 0
5* 2. 0	10.0
5. 0 -2	3. 0
5. 0 / 2. 0	2.5

38

عملگرهای افزایش و کاهش

در C++ ، افزایش یک واحد به مقدار یک متغیر از نوع صحیح را افزایش و بطور مشابه کاهش یک واحد از مقدار یک متغیر از نوع صحیح را کاهش می‌نامند.

عملگر کاهش را با – – و عملگر افزایش را با ++ نمایش می‌دهند. چون عملگرهای ++ و – – فقط روی یک عملوند اثر دارند این دو عملگر نیز جزء عملگرهای یکتائی می‌باشند.

39

مثال :

سه دستور العمل :

```
++x;
x++;
x=x+1;
```

معادل می‌باشند و بطریق مشابه سه دستورالعمل زیر نیز معادل می باشند.

```
- - y ;
y=y-1;
y- - ;
```

40

از عملگرهای ++ و -- می‌توان بدو صورت پیشوندی و پسوندی استفاده نمود.
در دستورالعمل‌های پیچیده عملگر پیشوندی قبل از انتساب ارزیابی میشود و عملگر پسوندی بعد از انتساب ارزیابی می‌شود.

مثال :

```
int x=5;
y=++x * 2;
```

پس از اجرای دستورالعملهای فوق :

```
y=12
```

```
int  x=5;
y=x++ * 2;
```

پس از اجرای دستورالعملهای فوق :

```
y=12
```

41

عملگر sizeof

Sizeof ازعملگرهای یکتائی می باشد و مشخص کننده تعداد بایت هائی است که یک نوع داده اشغال می‌کند.

مثال :

```
int  x;
cout << sizeof x ;
```

مقدار ۲ نمایش داده می‌شود .

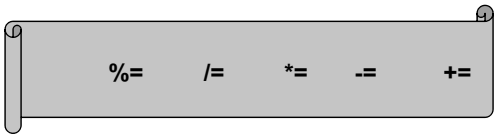
```
cout << sizeof(float) ;
```

مقدار ۴ نمایش داده می شود.

42

عملگرهای جایگزینی محاسباتی

برای ساده‌تر نوشتن عبارتها در C++ ، می‌توان از عملگرهای جایگزینی محاسبـــــاتی استفاده نمود.



مثال :

y+=x ---- → y=y+x
y-=x -----→ y=y-x
y*=x ----→ y=y*x

اولویت عملگرها

ارزیابی مقدار یک عبارت ریاضی براساس جدول اولویت عملگرها انجام می‌گردد. در ذیل جدول اولویت عملگرها براساس بترتیب از بیشترین اولویت به کمترین اولویت داده شده است.

چپ به راست	پرانتزها	( )
راست به چپ	عملگرهای یکتایی	sizeof ++ -- + -
چپ به راست	عملگرهای ضرب و تقسیم و باقیمانده	* / %
چپ به راست	عملگرهای جمع و تفریق	+ –
چپ به راست	عملگرهای درج و استخراج	>> <<
راست به چپ	عملگرهای جایگزینی و انتساب	= += -= *= /= %=

مهم

مثال ۱:

```
(5+2) *(6+2*2)/2
```

با توجه به جدول اولویت عملگرها داریم که

```
7 *(6+2*2)/2
7*(6+4)/2
7* 10 /2
70 /2
35
```

مثال ۲:

```
int a=6 , b=2, c=8, d=12;
d=a++ * b/c ++;
cout << d << c << b << a;
```

نتیجه

1 9 2 7

45

تمرین ۲ مهم HW2

1. مثال شماره ۲ را به صورت برنامه نویسی اجرا و نتایج را تحقیق نمایید.
2. در دستور زیر مقدار X پس از اجرای دستور چقدر است؟

```
X=(3*9*(3+(9*3/(3))));
```

3. کد زیر چه چیزی را چاپ می کند؟

```
cout<< "*\n**\n***\n****\n*****\n"
```

4. برنامه ای بنویسید که پنج عدد صحیح را بخواند و بزرگترین و کوچکترین آنها را بدست آورد و چاپ کند.
5. برنامه ای بنویسید که یک عدد صحیح را بخواند و مشخص کند این عدد فرد است یا زوج؟
6. برنامه ای بنویسید که دو عدد صحیح را بخواند و مشخص کند که آیا اولی مضربی از دومی است یا خیر ؟
7. برنامه ای بنویسید که توان ۲ و توان ۳ اعداد یک تا ۱۰ را به صورت یک جدول چاپ کند.

46

توضیحات (Comments)

توضیحات در برنامه باعث خوانائی بیشتر و درک بهتر برنامه میشود. بنابراین توصیه بر آن است که حتی الامکان در برنامه‌ها از توضیحات استفاده نمائیم. در C++، توضیحات بدو صورت انجام می‌گیرد که در اسلایدهای بعد به آن اشاره شده است.

الف: این نوع توضیح بوسیله // انجام می‌شود. که کامپیوتر هر چیزی را که بعد از // قرار داده شود تا انتهای آن خط اغماض می‌نماید.

مثال :

```
c=a+b; //c is equal to sum of a  and b
```

ب:توضیح نوع دوم با */ شروع شده و به /* ختم می‌شود و هر چیزی که بین* / و /* قرار گیرد اغماض می‌نماید .

مثال :

```
/ * this is a program
to calcufate sum of
n integer numbers */
```

47

توابع کتابخانه و نحوه فراخوانی آن

زبان C++ مجهز به تعدادی توابع کتابخانه می‌باشد. بعنوان مثال تعدادی توابع کتابخانه برای عملیات ورودی و خروجی وجود دارند. معمولاً توابع کتابخانه مشابه ، بصورت برنامه‌های هدف (برنامه ترجمه شده بزبان ماشین) در قالب فایلهای کتابخانه دسته بندی و مورد استفاده قرا رمی‌گیرند. این فایلها را **فایلهای header** می‌نامند و دارای پسوند **.h** می‌باشند.

برای استفاده از توابع کتابخانه خاصی بایستی نام فایل header آنرا در ابتدای برنامه در دستور #include قرار دهیم.

#include < اسم فایل header >

48

تعدادی از فایلهای Header

فایل هیدر	شرح	نوع	تابع
stdlib.h	قدرمطلق i	*int*	abs(i)
math.h	کسینوس d	*double*	cos(d)
math.h	e^x	*double*	exp(d)
math.h	$\log_e d$	*double*	log(d)
math.h	$\text{Log}_{10} d$	*double*	log10(d)
math.h	سینوس d	*double*	sin(d)
math.h	جذر d	*double*	sqrt(d)
string.h	تعداد کرکترهای رشته s	*int*	strlen(s)
math.h	تانژانت d	*double*	tan(d)
stdlib.h	کداسکی کرکتر c	*int*	toascii(c)
stdlib.h	تبدیل به حروف کوچک	*int*	tolower(c)
stdlib.h	تبدیل به حرف بزرگ	*int*	toupper(c)

برنامه در C++

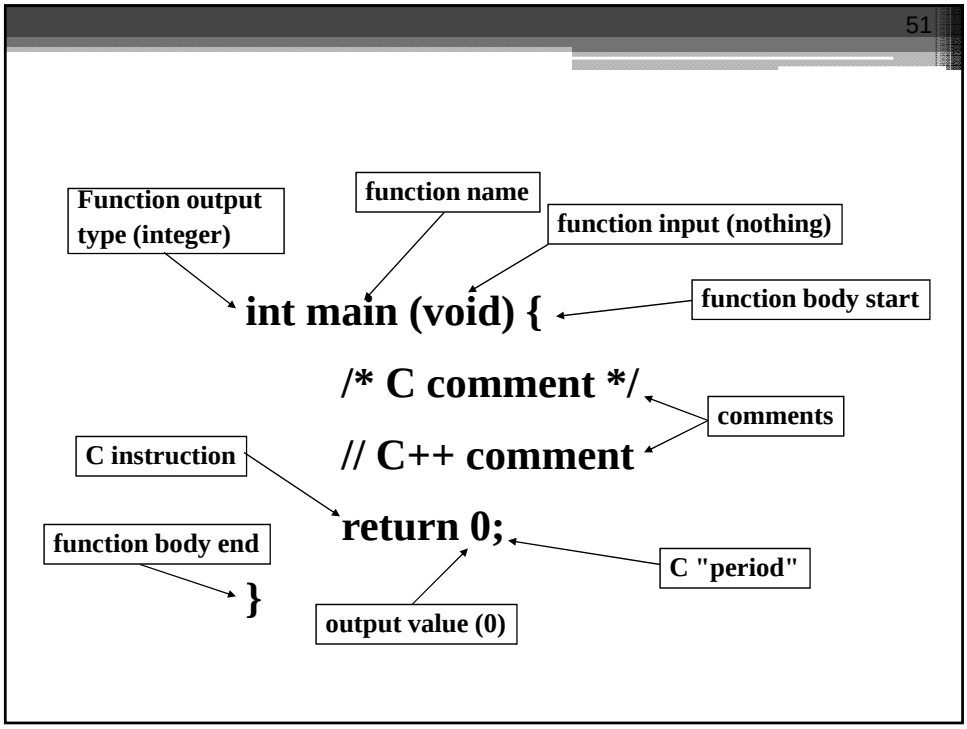
اکنون باتوجه به مطالب گفته شده قادر خواهیم بود که تعدادی برنامه ساده و کوچک به زبان C++ بنویسیم. برای نوشتن برنامه بایستی دستورالعملها را در تابع **main( )** قرار دهیم و برای اینکار می‌توان به یکی از دو طریقی کـه در اسلایدهای بعد آمده است ، عمل نمود.

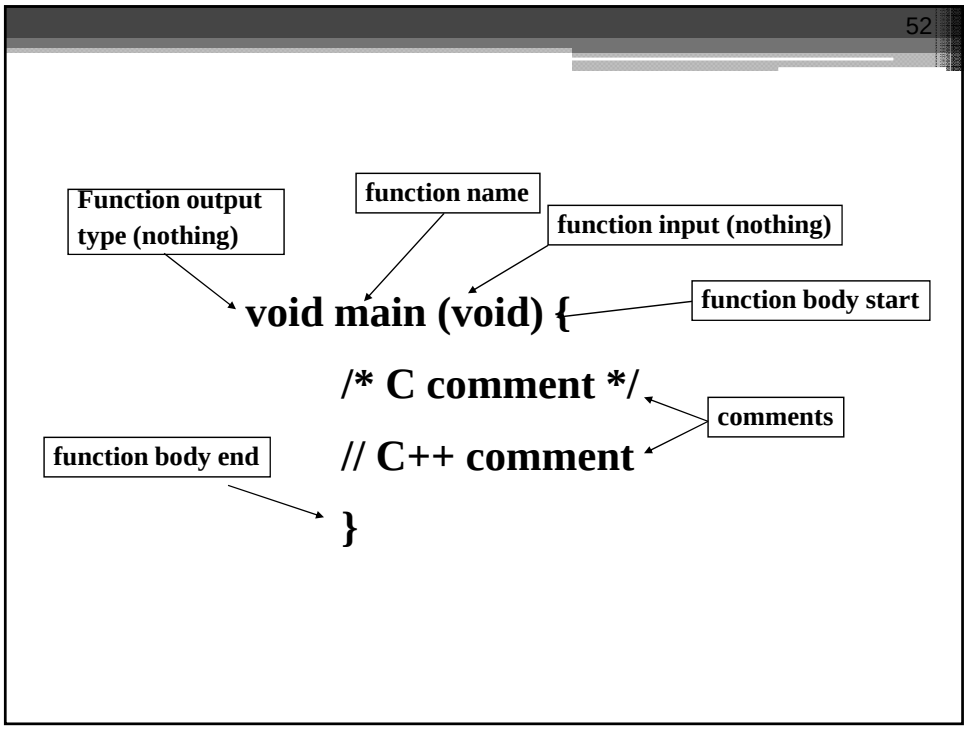
روش اول:

```
#include    <    >
int main( )
{
; دستورالعمل ۱
; دستورالعمل ۲
.
.
.
n دستورالعمل;
return 0 ;
}
```

روش دوم:

```
#include    <    >
void main( )
{
; دستورالعمل ۱
; دستورالعمل ۲
.
.
.
n دستورالعمل;
}
```


51
Function output type (integer)
function name
function input (nothing)
int main (void) {
function body start
/* C comment */
// C++ comment
comments
C instruction
return 0;
C "period"
function body end
}
output value (0)


52
Function output type (nothing)
function name
function input (nothing)
void main (void) {
function body start
/* C comment */
// C++ comment
comments
function body end
}

53

نکات

- error: به خطاهای برنامه نویسی error می گویند.
- انواع خطاها در برنامه نویسی:
  - خطاهای زمان compile(compile errors):
    - مانع کامپایل صحیح برنامه می شوند.
  - خطاهای زمان link (Link errors):
    - برای کامپایل مزاحمتی ایجاد نمی کنند اما مانع Link برنامه می شوند.
  - خطاهای زمان اجرا: (Run time errors):
    - کامپایل و Link با موفقیت انجام می شود ولی اجرای برنامه دچار اشکال می شود .

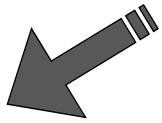
54

error

- حسن سیب را خورد.
- هسن سیب را خورد.
  - متناظر با خطای کامپایل
- را حسن خورد سیب.
  - متناظر با خطای Link
- سیب حسن را خورد.
  - متناظر با خطای زمان اجرا

55

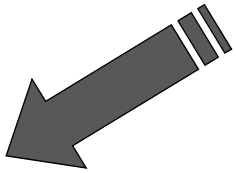
برنامه ای که پیغام C++ is an object oriented language را روی صفحه مانیتور نمایش می دهد.



```
#include   <iostream.h>
int main( )
{
cout <<"C++ is an object oriented language \n" ;
return 0 ;
}
```

56

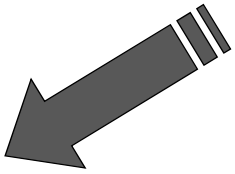
برنامه زیر یک حرف انگلیسی کوچک را گرفته به حرف بزرگ تبدیل می‌نماید.



```
#include   <iostream.h>
#include   <stdlib. h>
int main( )
{
 char c1 , c2;
cout << "Enter a lowercase letter:"
cin >> c1;
c2 = toupper(c1);
cout << c2 << endl;
return 0; }
```

57

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



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

58

فصل دوم

ساختارهای تصمیم گیری و تکرار

فهرست مطالب فصل دوم

عملگرهای رابطه ای

از این عملگرها برای تعیین اینکه آیا دو عدد با هم معادلند یا یکی از دیگری بزرگتر یا کوچکتر می‌باشد استفاده می‌گردد. عملگرهای رابطه‌ای عبارتند از:

مساوی	= =
مخالف	! =
بزرگتر	>
بزرگتر یا مساوی	> =
کوچکتر	<
کوچکتر یا مساوی	<=

61

عملگر شرطی

شکل کلی عملگر شرطی بصورت زیر می‌باشد:

```
expression _ test ? expression _ true : expression _ false
```

عملگر شرطی تنها عملگری در C++ می‌باشد که دارای سه عملوند می‌باشد.

یعنی:

*****?*****:*****

62

مثال ۱ :

```
int x=10, y=20, b;
B = (x>y) ? x : y ;
```

این دو دستور العمل باعث میشوند که ماکزیمم مقادیر y و x در b قرار بگیرد.

مثال ۲ :

```
x>=10 ? cout << "passed" : cout << "failed" ;
```

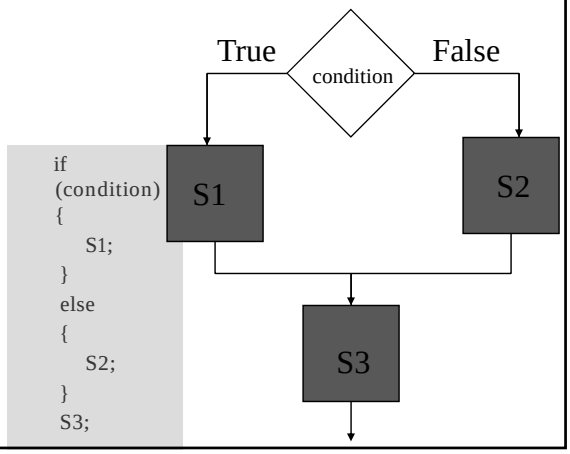
اگر مقدار x بزرگتر یا مساوی ده باشد رشتهٔ passed در غیر اینصورت رشته failedنمایش داده میشود.

63

دستورالعمل شرطی IF

توسط این دستور شرطی را تست نموده و بسته به آنکه شرط درست یـا غلط باشد عکس‌العمل خاصی را نشان دهیم.

```
if (عبارت )
{
    دستورالعمل 1 ;
       …
    دستورالعمل n;
}
 else
{
    دستورالعمل 1 ;
       …
    دستورالعمل n;
}
```




```
if
(condition)
{
   S1;
}
else
{
   S2;
}
S3;
```

64

مثال ۱ :

```
if(x != y)
{
  cout << x ;
  ++ x ;
}

else
{
  cout << y ;
  y ;
}
```

65

مثال ۲:

برنامه زیر یک عدد اعشاری را از ورودی گرفته جذر آنرا محاسبه می‌نماید.

```
#include <iostream.h>
#include <math . h>
int main( )
{
float  x,s;
cin >> x ;
if( x < 0 )
cout << " x is negative" << endl ;
else
{
s = sqrt(x) ;
cout << s << endl ;
          }
return 0;
}
```

66

عملگر کاما

تعدادی عبارت را می‌توان با کاما بهم متصل نمود و تشکیل یک عبارت پیچیده‌تری را داد. این عبارتها به ترتیب از چپ به راست ارزیابی شده و مقدار عبارت معادل عبارت **n** می‌باشد.

(عبارت n , ... , عبارت 3 , عبارت 2 , عبارت 1)

67

مثال :

اگر داشته باشیم **int a=2 , b=4 , c=5 ;** **عبارت زیر را در نظر بگیرید:**

(++ a , a+b, ++ c, c+b)

مقدار عبارت برابر است با b+c که معادل 10 می‌باشد.

68

عملگرهای منطقی

با استفاده از عملگرهای منطقی می‌توان شرطهای ترکیبی در برنامه ایجاد نمود.
عملگرهای منطقی عبارتست از :

AND
OR
NOT

که در C++ به ترتیب بصورت زیر نشان داده میشود.

&&
||
!

جدول درستی سه عملگر شرطی

And

a	b	a && b
true	true	True
true	false	False
false	true	False
false	false	False

a	b	a && b
1	1	1
1	0	0
0	1	0
0	0	0

Or

a	b	a || b
true	true	True
true	false	True
false	true	True
false	false	False

a	b	a || b
1	1	1
1	0	1
0	1	1
0	0	0

Not

a	!a
true	False
false	True

چند مثال :

```
if ((x= = 5) ||(y != 0))
   cout << x << endl ;
```

اگر x برابر با 5 یا y مخالف صفر باشد مقدار x نمایش داده شود .

```
if(!x)
 x = 0 ;
```

اگر مقدار x مخالف صفر باشد، آنگاه x برابر با صفر شود .

71

برنامه زیر طول سه پاره‌خط را از ورودی گرفته مشخص می‌نماید که آیا تشکیل یک مثلث میدهد یا خیر؟

```
#include < iostream.h >
int main( )
{
float  a, b, c;
cout << "Enter three real numbers" << endl ;
cin >> a >> b >> c;
if(( a < b + c) &&(b < a+c) &&(c < a+b))
cout << "It is a triangle" ; // Yani yek 3 zelie ast
else
cout << "Not a triangle" ;  // Yani yek 3 zelie nist
return  0 ;
}
```

تمرین ۳ HW3

72

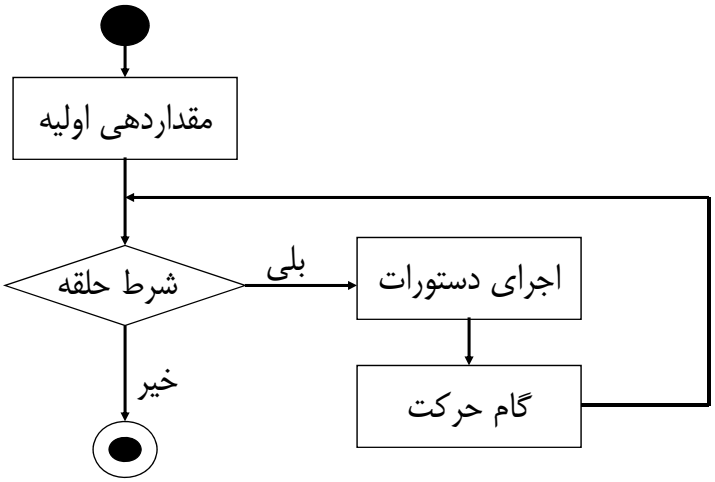
دستورالعمل For

از دستور العمل **for** برای تکرار دستورالعملها استفاده میشود. شکل کلی دستور **for** بصورت زیر می‌باشد:

```
for (عبارت 1 ; عبارت 2 ; عبارت 3)
 {
  دستورالعمل 1 ;
  دستورالعمل 2 ;
  .
  .
  .
  دستورالعمل n;
 }
```

73

الگوریتم اجرای for



74

برنامه زیر عدد صحیح و مثبت n را از ورودی گرفته فاکتوریل آنرا محاسبه ونمایش می‌دهد.

```
#include <iostream.h>
int     main( )
{
int  n, i ;
long fact = 1 ;
cout << "Enter a positive integer number";
cin >> n;
for( i=1; i<=n; ++i)
   fact *= i;
 cout << fact << endl;
return 0 ;
}
```

75

برنامه زیر ارقام 0 تا 9 را نمایش می‌دهد.

```
#include   <iostream.h>
int main( )
{
int j=0 ;
for( ; j <= 9 ; )
   cout << j++ << endl;
return 0 ;
}
```

HW3 تمرین ۳

76

برنامه زیر مجموع اعداد صحیح و متوالی بین ۱ تا n را محاسبه نموده و نمایش می‌دهد.

```
#include <iostream.h>
int    main( )
{
int n, i=1 ;
long s = 0 ;
cin >> n ;
for(; i<=n; i++)
   s += i;
cout << s ;
return  0 ; }
```

HW3 تمرین ۳

77

برنامه زیر کلیه اعداد سه رقمی که با ارقام 1 ، 2 ، 3 ایجاد می‌شوند را نمایش می‌دهد.

```
#include   <iostream.h>
int main( )
{
int i,j,k,n;
for(i=1; i<=3; ++i)
  for(j=1; j<=3; ++j)
       for(k=1; k<=3; ++k)
       {
       n=i*100 + j*10+k;
       cout << n << '\n' ;
       }
return 0 ;
}
```

HW3 تمرین ۳

78

HW3 تمرین ۳

1. کد برنامه های تمرینهای اسلایدهای ۷۱، ۷۵، ۷۶ و ۷۷ را نوشته و خروجی آن را به عنوان خروجی برنامه ارسال کنید.
2. برنامه ای بنویسید که یک عدد صحیح را پرسیده و تمام مقادیر بین این اعداد (و خروجی آن) تا عددی که ۱۰ واحد بزرگتر از آن باشد چاپ نماید. (بدین معنا که اگر ورودی عدد ۵ باشد خروجی اعداد ۵ تا ۱۵ را چاپ نماید.)
3. برنامه ای بنویسید که از شما عددی معادل با تعداد روز را بپرسد و سپس این مقدار را برحسب هفته و روز بدست آورد. برای مثال باید ۱۸ روز را به ۲ هفته و ۴ روز تبدیل نماید.

79

فصل سوم

سایر ساختارهای تکرار

80

فهرست مطالب فصل سوم

دستورالعمل while

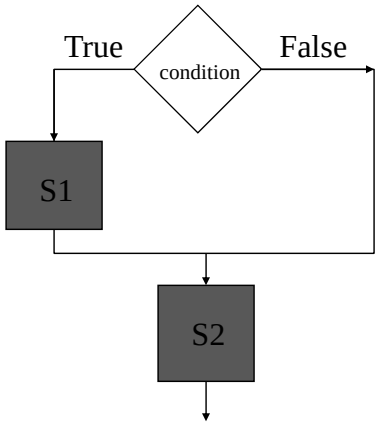
از این دستور العمل مانند دستورالعمل **for** برای تکرار یک دستورالعمل ساده یا ترکیبی استفاده می‌گردد. شکل کلی این دستور العمل بصورت زیر می‌باشد.

```
while( شرط)
{
 ; دستورالعمل ۱
 ; دستورالعمل ۲
    .
    .
 ; دستورالعمل n
}
```



```
while (condition)
  {
    S1;
  } {
  S2;
```

```
while( شرط)
{
 ; دستورالعمل ۱
 ; دستورالعمل ۲
    .
    .
 ; دستورالعمل n
}
```



تمرین شماره ۴

۱- خروجی برنامه زیر را تحقیق نمایید. نوشتن و ارسال برنامه الزامی است

```
#include <iostream.h>
void main()
{
   int a,b,c;
   cin >> a >> b >> c;
   if (a <=b)
   {
      cout << “min is “ << a << endl;
    }
   else
   {
      cout << “ min is “ << b << endl;
   }
   cout << “happy now?” << endl;
}
```


تمرین شماره ۴

۲- خروجی برنامه زیر را تحقیق نمایید. نوشتن و ارسال برنامه الزامی است

```
//read 100 numbers from the user and output their sum
#include <iostream.h>
void main()
{
   int i, sum, x;
   sum=0;
   i=1;
   while (i <= 100)
   {
         cin >> x;
         sum = sum + x;
         i = i+1;
   }
cout << “sum is “ << sum << endl;
}
```

85

تمرین شماره ۴

۳- برنامه ای بنویسید که از کاربر بپرسد
"آیا مایل هستید که برنامه خاتمه یابد؟ (Y/N)"

Do you want to use this program? (Y/N)

اگر کاربر با Y جواب دهد برنامه خاتمه یابد و بنویسد
"برنامه بسته شد."

the program terminated

اما اگر با N جواب دهد
"آیا واقعاً مایل هستید که از برنامه خارج نشوید؟ (Y/N)"

Are you really sure you do not want to use this program? (Y/N)

اگر کاربر با N جواب دهد برنامه تمام شود در غیر این صورت بنویسد
"آیا واقعاً واقعاً مایل هستید که از برنامه خارج نشوید؟ (Y/N)"

Are you really really sure you do not want to use this program? (Y/N)

و به همین صورت هر دفعه یک واقعاً (really) به برنامه اضافه نماید.

86

تفاوت دستورهای for وwhile

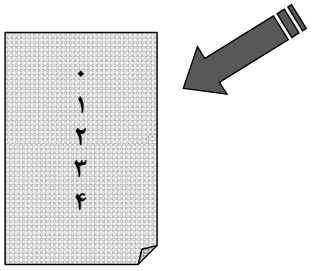
دستورالعمل **for** زمانی استفاده میشود که تعداد دفعات تکرار از قبل مشخص و معین باشد. در صورتیکه تعداد دفعات تکرار مشخص نباشد بایستی از دستورالعمل **while** استفاده نمود.

87

مثال :

```
int x=0
while(x<5)
cout << x ++<< endl;
```

با اجرای قطعه برنامه فوق مقادیر زیر نمایش داده میشود :



88

endl چیست؟

این دستور به معنی End line یعنی برای رفتن به سطر بعدی هستش که عمل Enterرو برای شما انجام میده.

به طور مثال دستور زیر:

Cout<<"ashiyane"<<endl

بعد از چاپ عبارت یک سطر به پایین خواهد رفت

دستور زیر معادل دستور بالا هستش با قرار دادن عبارت \n عمل Enterاعمال می شود.

Cout<<"ashiyane \n"

89

برنامهٔ فوق n مقدار از نوع اعشاری را گرفته میانگین آنها را محاسبه و در متغیر avg قرار می‌دهد.

```
#include <iostream.h>
int main( )
{
int count = 0 , n;
float x, sum = 0 , avg ;
cin >> n ; /* تعداد مقادیر ورودی n*/
while(count < n){
cin >> x ;
sum += x ;
++ count ; }
avg = sum / n ;
cout << avg << endl;
return 0 ; }
```

90

دستورالعمل do while

این دستور العمل نیز برای تکرار یک دستورالعمل ساده یا ترکیبی استفاده می‌شود. شکل کلی این دستورالعمل بصورت زیر می‌باشد.

```
do
{
 ; دستورالعمل ۱
 ; دستورالعمل ۲
    .
    .
 ; دستورالعمل n
} while( شرط);
```

91

تفاوت دستورهای do while و while

در دستورالعمل **while** ابتدا مقدار شرط ارزیابی شده اما در دستورالعمل **do while** ابتدا دستورالعمل اجرا شده سپس مقدار شرط ارزیابی می‌گردد. بنابراین دستورالعمل **do while** حداقل یک بار انجام میشود .

92

مثال :

```
#include   <iostream.h>
int main( )
{
int count = 0;
do
cout << count ++<<endl ;
while(count <= 9);
return 0 ; }
```

ارقام **0** تا **9** را روی ده خط نمایش می‌دهد

93

دستورالعمل break

این دستورالعمل باعث توقف دستورالعملهای تکرار (for , while ,do while) شده و کنترل به خارج از این دستورالعملها منتقل می‌نماید.

Break

94

مثال ۱:

```
#include  <iostream.h>

int main( )
{
float x, s=0.0 ;
cin >> x ;
while(x <= 1000.0) {
if(x < 0.0){
cout << "Error-Negative Value" ;
break;
}
s += x ;
cin >> x ;}
cout << s << endl ;
return 0 ; }
```

95

مثال ۲:

```
#include  <iostream.h>
int main( )
{
int count = 0 ;
while( 1 )
{
count ++ ;
if(count > 10 )
break ;
}
cout << "counter :  " << count << "\n";
return 0 ;
}
```

96

مثال ۳:

```
#include  <iostream.h>
void  main( )
{
int count;
float x, sum = 0;
cin >> x ;
for(count = 1; x < 1000 . 0; ++ count )
{
cin >> x ;
if(x < 0.0) {
cout << "Error – Negative value " <<endl;
break ;
}
sum += x ; }
cout << sum << '\n' ; }
```

مثال ۴:

```
#include  <iostream.h>
int main( )
{
float x , sum = 0.0 ;
do
{
cin >> x ;
if(x < 0.0)
   {
        cout << "Error – Negative Value" << endl ;
        break ;
   }
   sum += x ;
} while(x <= 1000.0);
cout << sum << endl ;
return 0 ;
}
```


دستورالعمل continue

از دستورالعمل **continue** می‌توان در دستورالعملهای تکرار **for ، while ، do while** استفاده نمود. این دستورالعمل باعث می‌شود که کنترل بابتدای دستورالعملهای تکرار منتقل گردد.

Continue

99

مثال ۱:

```
#include  <iostream.h>
int main( )
{
float  x, sum = 0.0 ;
Do
 {
cin >> x ;
if(x < 0 . 0)
   {
cout  << "Error" << endl ;
continue ;
   }
sum += x ;
} while(x <= 1000.0 );
cout << sum ;
return 0 ;
}
```

100

مثال ۲:

```
#include  <iostream.h>
int main( )
{
int n , navg = 0;
float  x, avg, sum = 0;
cin >> n; / * عبارت از تعداد اعداد ورودي n * /
for (int count = 1 ; count <=n; ++ count )
{
        cin >> x;
        if(x < 0 )
                continue;
        sum += x;
        ++ navg;
}
avg = sum / navg;
cout << avg << endl;
return 0;
 }
```

101

دستورالعمل switch

همانطور که می دانید از دستورالعمل شرطی(**if else**) می توان بصورت تودرتو استفاده نمود ولی از طرفی اگر عمق استفادهٔ تو در تو از این دستورالعمل زیاد گردد، درک آنها مشکل میشود . برای حل این مشکل **C++** ، دستورالعمل **switch** که عملاً یک دستورالعمل چند انتخابی می‌باشد را ارائه نموده است.

switch case

102

شکل کلی دستور العمل Switch

```
switch(عبارت)
{
case    valueone : statement;
                   break;
case     valuetwo: statement;
                   break;

case      valuen : statement;
                  break;
default: statement ;
}
```

103

شکل کلی دستور العمل Switch

```
switch (عبارت) {
  case مقدار ۱ :
                    دستورات ۱
                    break ;
  case مقدار ۲ :
                    دستورات ۲
                    break ;
      .
      .
      .
  default :
                    n دستورات
}
```

104

مثال ۱ :

```
#include <iostream.h>
void main( )
{
unsigned int n ;
cin >> n;
switch(n)
{
  case 0:
       cout << "ZERO" << endl ;
       break;
  case 1:
       cout << "one" << endl ;
       break ;
  case 2:
       cout << "two" << endl ;
       break;
  default :
       cout << "default" << endl;
}     / * end of switch statement * /
}
```

105

مثال ۲ :

```
#include  <iostream.h>
void main( )
{
unsigned int  n;
cin >> n ;
switch(n) {
case 0 :
case 1:
case 2:
      cout << "Less Than Three" << endl;
       break;
case 3:
       cout << "Equal To Three" << endl ;
        break;
default:
       cout << "Greater Than Three" << endl;
         }
}
```

106

مثال ۳ :

```
char ch;
switch (ch) {
  case '+' :
              r = x + y ;
              break ;
  case '-' :
              r = x - y ;
              break ;
  case '*' :
              r = x * y ;
              break ;
  case '-' :
              r = x - y ;
              break ;
  case '*' :
              r = x * y ;
              break ;
  case '/' :
              r = x / y ;
              break ;
  default :
              r = 0 ;
              printf
("Invalid operator.") ;
}
```

107

تابع cin.get() :

108

قطعه برنامه ذیل یک کاراکتر را از صفحه کلید گرفته و نمایش می‌دهد.

```
char     x;
x = cin.get( );
cout << x ;
```

برنامهٔ ذیل یک سطر متن انگلیسی که به CTRL Z ختم میشود را گرفته دقیقاً نمایش می‌دهد.

```
#include <iostream.h>
int main( )
{
char x;
while((x = cin.get( ) !=EOF))
cout << x ;
return 0 ;
 }
```

EOF به معنی End of File می‌باشد که در iostream.h تعریف شده و مقدار آن برابر با ۱- می‌باشد. مقدار آن در سیستم عامل DOS عبارتست از ctrl z .

در قطعه برنامه ذیل ازتابع cin.get() و دستور switch استفاده شده است.

```
char      x;
x = cin.get( );
switch(x) {
case 'r':
case 'R':
        cout << "RED" << "\n" ;
        break ;
case 'b':
case 'B':
         cout << "BLUE" << endl ;
         break ;
case 'y':
case 'Y':
        cout <<  "YELLOW" << endl;
  }
```

111

برنامهٔ ذیل یک سطر متن انگلیسی را گرفته کاراکترهای خالی (blank) آنرا حذف نموده و نمایش میدهد.

```
#include <iostream.h>
int main( )
{
char next;
while((next = cin.get( ) ) !=EOF)
if(next != ' ')
cout  << next ;
return 0 ;
 }
```

112

عملگر static_cast

از این عملگر برای تبدیل موقت یک نوع data به نوع دیگر استفاده می‌شود. این عملگر یک عملگر یکتائی می‌باشد.

113

مثال ۱:

```
int   x = 25 ;
float  y ;
y = static_cast
< float >(x) ;
```

مقدار x موقتاً بصورت اعشـــاری در می آید و در نتیجه مقدار y برابر بـا 25.0 می شود. بایستی توجه داشت که نوع متغیر x عوض نمی شود بلکه موقتاً مقدار آن بصورت اعشـــاری در آمده است.

114

مثال ۲:

```
float x = 14.75 ;
cout << static_cast
< int >(x) << endl;
cout << x ;
```

ابتدا مقدار ۱۴ نمایش داده میشـــود و ســـپس مقـــدار ۱۴.۷۵ نمـــایش داده میشود.

جدول اولویت عملگرها

چپ به راست	( )
راست به چپ	sizeof - + -- ++ ( ) Static_cast < >
چپ به راست	* / %
چپ به راست	- +
چپ به راست	<< >>
چپ به راست	< <= > >=
چپ به راست	!= = =
راست به چپ	? :
راست به چپ	= += -= *= /= %=
چپ به راست	,

تمرین ۴ HW4

موعد تحویل: ۱۳۹۲/۰۸/۲۱

1. **کد برنامه های تمرینهای اسلایدهای ۸۳، ۸۴، ۹۴، ۹۵، ۹۷، ۱۰۰ و ۱۰۶ را بازنویسی نموده و خروجی آنها را به گزارش کار ارسال نمایید.**
2. **برنامه ای اسلاید ۸۵ را پیاده سازی و توضیح دهید.**

117

فصل چهارم

اعداد تصادفی

118

فهرست مطالب فصل چهارم

1. تولید اعداد تصادفی
2. تعریف نوع داده (**typedef**)
3. داده های از نوع شمارشی
4. فرمت های مختلفه مقادیر خروجی

اعداد تصادفی

مقادیر تصادفی یا شانسی در اکثر برنامه‌های کاربردی در زمینه شبیه سازی و بازیهای کامپیوتری نقش مهمی را ایفا می‌نمایند. بـرای ایجـاد یک عدد تصادفی صحیح بـین ۰ و ۳۲۷۶۷ بایسـتی از تـابع **rand()** استفاده نمائیم.

rand( )

برنامه زیر 10 عدد تصادفی بین 0 و 32767 را ایجاد می‌نماید.

```
#include      <stdlib.h>
#include      < iostream.h>
int main( )
{
for(int j=1; j<=10; ++j)
cout << rand( ) << '\n';
return 0 ;
}
```

نکته :

اگر برنامهٔ فوق را چندبار اجرا نمائیم جواب یکسانی را از کامپیوتری‌می گیریم.
برای تصادفی کردن اعداد می‌بایستی از تابع ( )srand استفاده نمائیم.
این تابع به یک آرگومان صحیح از نوع unsigned نیاز دارد.
به این آرگومان seed گفته می‌شود.

برنامه زیر 10 عدد تصادفی بین 0 و 32767 را ایجاد می‌نماید. (( )srand)

```
#include <stdlib.h>
#include < iostream.h>
int main( )
{
unsigned   seed;
cout << "Enter seed value : " ;
cin >> seed ;
srand(seed);
for(int j=1; j<=10; ++j)
cout << rand( ) << ' \n ';
return 0 ;
 }
```


برنامه زیر نتیجه پرتاب دو تاس را نمایش می‌دهد.

```
#include       < iostream.h>
#include       <stdlib.h>
int main( )
{
unsigned   seed, d1, d2;
cout << "Enter seed: " ;
cin >> seed ;
srand(seed) ;
d1= 1+rand( )% 6 ;
d2= 1+rand( )% 6 ;
cout << d1 << "      "<< d2 ;
return 0 ;
 }
```

123

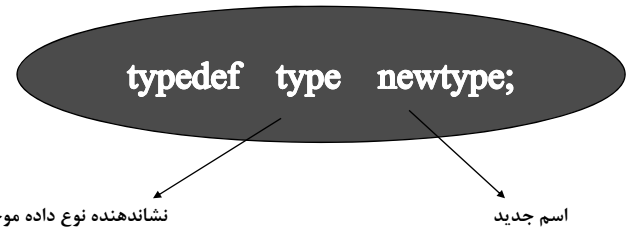
برنامه زیر 10 اعداد شانسی بین 0 و ۱ را نمایش می‌دهد.

```
#include      <stdlib.h>
#include      < iostream.h>
int main( )
{
unsigned seed ;
cout << "Enter seed: " ;
cin >> seed ;
srand (seed) ;
for (int i=1; i<=10; ++i)
cout << rand( ) / 32768.0 << endl ;
return 0 ;
}
```

124

تعریف نوع داده (typedef)

از **typedef** می‌توان برای تعریف نوع داده‌های جدید که معادل نوع داده‌های موجود باشد استفاده نمود. شکل کلی عبارتست از :



مثال

```
typedef int integer;
```

حال می‌توان **y** و **x** را بصورت زیر تعریف نمود :

```
integer x,y;
```

125

داده‌های از نوع شمارشی

بمنظور معرفی داده‌های از نوع شمارشی از کلمه **enum** استفاده می‌گردد.

مثال :

color یک نوع داده شمارشی می‌باشد.

126

چند مثال :

```
emum status {married, devorced, vidow, single};
status  a ;
a= single ;
```

```
enum days {sat, sun, mon, tue, wed, thr, fri};
```

```
enum bread {lavash, fantezi, taftoon, barbari};
```

```
enum color { yellow, red=2, brown, white };
color  x=brown;
```

0 3 4

توجه :

بایستی در نظر داشت که داده‌های از نوع شمارشی در عملیات ورودی و خروجی شرکت نمی‌نمایند. بعبارت دیگر مقادیر داده‌های از نوع شمارشی بایستی در برنامه تعیین نمود. دستورالعلهای ورودی و خروجی مانند cin و cout در مورد داده‌های شمارشی نمی‌توان استفاده نمود.

فرمتهای مختلف مقادیر خروجی

مقدار x بطور غیر علمی با نقطه اعشار ثابت نمایش داده می شود.

```
include <iomanip.h>

double  x=1050 ;
cout << setiosflags(ios : : fixed | ios: : showpoint ) <<
   setw(23) << setprecision(2) << x << endl ;
```

مقدار x با طول میدان ۲۳ نمایش داده می شود.

مقدار x با دو رقم اعشار نمایش داده می شود.

بنابراین مقدار x بصورت زیر نمایش داده می شود :

1050.00 شانزده ستون خالی

129

فصل پنجم

آرایه ها

130

فهرست مطالب فصل پنجم

1. آرایه یک بعدی
2. آرایه دو بعدی (ماتریس ها)

131

آرایه یک بعدی

آرایه یک فضای پیوسته از حافظه اصلی کامپیوتر می‌باشد که می‌تواند چندین مقدا را در خود جای دهد.

کلیه عناصر یک آرایه از یک نوع می‌باشند.

عناصر آرایه بوسیله اندیس آنها مشخص می‌شوند.

در C++ ، اندیس آرایه از صفر شروع می‌شود.

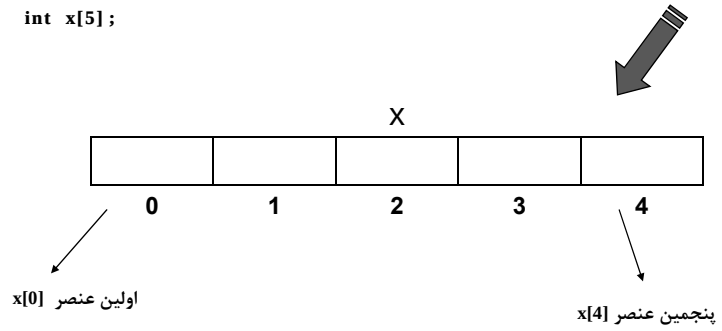
132

کاربرد آرایه ها

آرایه‌ها در برنامه‌نویسی در مواردی کاربرد دارند که بخواهیم اطلاعات و داده‌ها را در طول اجرای برنامه حفظ نمائیم.

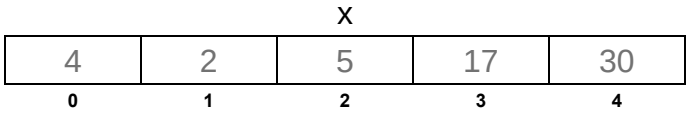
آرایه یک بعدی از نوع int

```
int x[5] ;
```



تخصیص مقادیر اولیه به عناصر آرایه :

```
int x[5]= {4, 2, 5, 17, 30};
```



دریافت مقادیر عناصر آرایه :

```
int x[5];
for(int   i=0; i<=4; ++i)
cin >> x[ i ] ;
```

نمایش مقادیر عناصر آرایه :

```
for(int   i=0; i<5; ++i) cout << x[ i ] ;
```

اگر تعداد مقادیر اولیه کمتر از تعداد عضوهای آرایه باشد عضوهای باقیمانده بطور اتوماتیک، مقدار اولیه صفر می‌گیرند.

```
int  x[5] = {12, 5, 7};
```

X

12	5	7	0	0
0	1	2	3	4

بایستی توجه داشت که آرایه‌ها به صورت ضمنی مقدار اولیه صفر نمی‌گیرند. برنامه نویس باید به عضو اول آرایه، مقدار اولیه صفر تخصیص دهد تا عضوهای باقی‌مانده بطور اتوماتیک، مقدار اولیه صفر بگیرند.

```
int  x[5] = {0} ;
```

X

0	0	0	0	0
0	1	2	3	4

دستور زیر یک آرایه یک بعدی شش عنصری از نوع float ایجاد می‌نماید.

```
float x[ ] = {2.4, 6.3, -17.1, 14.2, 5.9, 16.5} ;
```

X

2.4	6.3	-17.1	14.2	5.9	16.5
0	1	2	3	4	5

137

برنامه ذیل 100 عدد اعشاری و مثبت را گرفته تشکیل یک آرایه میدهد سپس مجموع عناصر آرایه را مشخص نموده نمایش می‌دهد.

```
#include <iostream.h>
#include <iomanip.h>
int main( )
{
   const int arrsize = 100 ;
   float x[ arrsize], tot = 0.0 ;
   for (int  j=0; j<arrsize; j++)
         cin >> x[ j ];
   for (j=0; j<arrsize; j++)
         cout << setiosflags(ios::fixed|ios :: showpoint ) << setw(12) <<
         setprecision(2) << x[ j ] << endl;
   for (j=0; j<arrsize; j++)
         tot += x[ j ] ;
cout << tot ;
return 0 ;
}
```

138

برنامه ذیل 20 عدد اعشاری را گرفته تشکیل یک آرایه داده سپس کوچکترین عنصر آرایه را مشخص و نمایش می‌دهد.

```
#include <iostream.h>
#include <conio.h>
int main( )
{
float  x[20], s;
int j ;
clrscr( ) ;
for(j=0; j<20 ; ++j) cin >> x[ j ];
s = x[0 ] ;
for(j=1; j<20; ++j)
if(x[ j] <s) s = x[ j ];
cout << s << endl;
return 0;
}
```

برنامه زیر 100 عدد اعشاری را گرفته بروش حبابی (Bubble sort) بصورت صعودی مرتب می‌نماید.

```
#include <iostream.h>
#include <conio.h>
int main ( )
{
float x[100] , temp;
int  i,j ;
clrscr( );
for(i=0; i<100; ++i)
   cin >> x[i ];
for(i=0; i<99; i++)
   for(j=i+1 ; j<100; j++)
                  if(x[ j ] < x[i ]
                  {
                           temp = x[ j ] ;
                           x[ j ] = x[ i ];
                           x[ i ] = temp ;
                  }
for(i=0; i<=99; i++)
   cout << x[ i ] << endl;
return 0 ;
}
```

آرایه‌های دوبعدی (ماتریس‌ها)

ماتریسها بوسیله آرایه‌های دوبعدی در کامپیوترنمایش داده میشوند.

int a[3][4];

	ستون 0	ستون 1	ستون 2	ستون 3
سطر 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
سطر 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
سطر 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

تخصیص مقادیر اولیه به عناصر آرایه :

```
int a[3][4]={ {1,2,3,4}, {5,6,7,8}, {9,10,11,12} } ;
```

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

نکته 1 :

```
int  a[3][4]= { {1}, {2,3} , {4,5,6} } ;
```

	0	1	2	3
0	1	0	0	0
1	2	3	0	0
2	4	5	6	0

نکته 2 :

```
int  a[3][4]= {1, 2, 3, 4,5 } ;
```

	0	1	2	3
0	1	2	3	4
1	5	0	0	0
2	0	0	0	0

نکته 3 :

در یک آرایهٔ دواندیسی، هر سطر، در حقیقت آرایه‌ای یک اندیسی است. در اعلان آرایه‌های دواندیسی ذکر تعداد ستونها الزامی است.

```
int  a[ ][4]={1,2,3,4,5};
```

	0	1	2	3
0	1	2	3	4
1	5	0	0	0

برنامه زیر یک ماتریس 4*3 را گرفته مجموع عناصر آن را مشخص نموده و نمایش می‌دهد.

```
#include <iostream.h>
#include <conio.h>
int main( )
{
float  x[3][4], total= 0.0;
int  i, j ;
// generate matrix x.
for(i=0; i<3; ++i)
for (j=0; j<4; j++)
cin >> x[ i ][ j ];
// calculate the sum of elements.
for(i=0; i<3; ++i)
for(j=0; j<4; j++)
total + = x [ i ][ j ];
cout << "total =  " << total << endl;
return 0 ;
}
```