برنامه سازی پیشرفته

مدرس:

جواد قهرمانی

نام درس :

برنامه سازی پیشرفته (رشته مهندسی فناوری اطلاعات)

تعداد واحد درسی :

۳ واحد

فهرست مطالب

فصل اول

مقدمات C++

فهرست مطالب فصل اول

تاریخچه مختصر C++

این زبان در اوائل دهه ۱۹۸۰ توسط 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.

قانون نامگذاری شناسه‌ها

۱) حروف کوچك و بزرگ در نامگذاري شناسه‌ها متفاوت مي‌باشند.

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

قانون نامگذاری شناسه‌ها

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

قانون نامگذاری شناسه‌ها

۳) 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

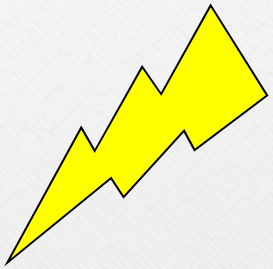
متغیرها

متغیـــر، مکـــانی در حافظـــه اصـــلی کامپیوتر می‌باشـد کـه در آنجـا یـک مقدار را می‌توان ذخیره و در برنامـه از آن اسـتفاده نمـود. قـانون نامگـذاری متغیرهـــا همـــان قـــانون نامگـــذاری شناسه‌ها می‌باشد.

انواع داده ها

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

اعلان متغیرها



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

چند مثال از اعلان متغیر ها :

- ✓ براي اعلان متغيرx از نوع int :

```
int   x;
```

- ✓ برای اعلان متغيرهاي p و q را از نوع float که هر کدام چهار بايت از حافظه را اشغال مي‌کنند :

```
float  p , q  ;
```

- ✓ برای اعلان متغير next از نوع کرکتر که مي‌توان يکي از ٢٥٦ کرکتر را به آن تخصيص داد و يك بايت را اشغال مي‌کند.

```
char next ;
```

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 .

- در دستورالعمل long a=67000 , b=260;

متغیرهاي b و a را از نوع long int تعریف نموده با مقادیر بترتیب 260 و 67000.

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.

کاراکترهای مخصوص

کامپیلر 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 کرکتر \a می‌توان برای ایجاد صدای beep استفاده نمود.

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

رشته‌ها

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

مثال ۱:

"BOOK STORE" یک رشته ده کرکتـری می‌باشـد که با توجه به کرکتر 0\ که به انتهای آن در حافظه اضافه می شود جمعـاً 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 کرکتر می‌باشد که یک بایت از حافظه را اشغال می‌نماید.

نمایش مقادیر داده‌ها

برای نمایش داده‌ها بر روی صفحه مانتور از cout که بدنبال آن عملگر درج یعنی << قید شده باشد استفاده می‌گردد. بایستی توجه داشت کـه دوکرکتـر < پشـت سر هم توسط C++ بصورت یک کرکتر 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 ;
```

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;
```

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.

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 26 میشود.

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 باشـد نتیجـه 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.

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

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..

عملگرهای افزایش و کاهش

عملگر کاهش را با – – و عملگر افزایش را بـــا ++ نمـــایش می‌دهنـــد. چـــون عملگرهـــای ++ و – – فقـــط روی یـــک عملوند اثر دارند این دو عملگر نیز جـــزء عملگرهای یکتائی می‌باشند.

مثال :

سه دستور العمل :

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

معادل مي‌باشند و بطريق مشابه سه دستورالعمل زير نيز معادل مي باشند.

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

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 اجراي دستورالعملهاي فوق :

```
y=12
```

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

پس از اجرای دستورالعملهای فوق :

```
y=14
```

sizeof **عملگر**

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

مثال :

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

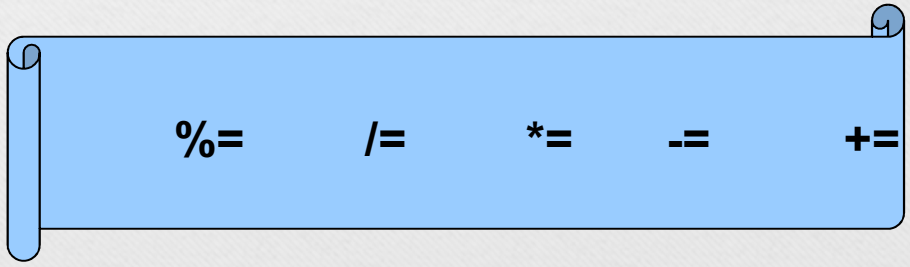
مقدار ۲ نمایش داده می‌شود .

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

مقدار ٤ نمایش داده مي شود.

عملگرهای جایگزینی محاسباتی

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



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	= += -= *= /= %=

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;
```

خروجی :

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.

توضیحات (Comments)

الف: اين نوع توضيح بوسيله // انجام مي‌شود. كه كامپيوتر هر چيزي را كه بعد از // قرار داده شود تا انتهاي آن خط اغماض مي‌نمايد.

مثال :

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

ب:توضيح نوع دوم با /* شروع شده و به */ ختم مي‌شود و هر چيزي كه بين* / و /* قرار گيرد اغماض مي‌نمايد .

مثال :

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

توابع کتابخانه

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

نحوه استفاده از توابع کتابخانه ای

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

#include < اسم فایل header >

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

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 که 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 ;
.
.
.
دستورالعمل n;
}
```

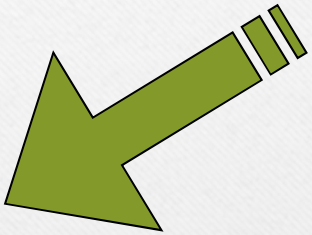
نكات–۲

- **error:** به خطاهاي برنامه نويسي error مي گويند.
- انواع خطاها در برنامه نويسي:
  - خطاهاي زمان compile(compile errors):
    - مانع كامپايل صحيح برنامه مي شوند.
  - خطاهاي زمان link (Link errors):
    - براي كامپايل مزاحمتي ايجاد نمي كنند اما مانع Link برنامه مي شوند.
  - خطاهاي زمان اجرا: (Run time errors):
    - كامپايل و Link با موفقيت انجام مي شود ولي اجراي برنامه دچار اشكال مي شود .

Error

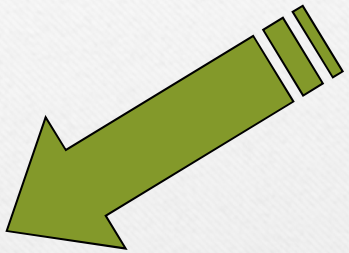
- حسن سیب را خورد.
- هسن سیب را خورد.
  - متناظر با خطاي كامپايل
- را حسن خورد سیب.
  - متناظر با خطاي Link
- سیب حسن را خورد.
  - متناظر با خطاي زمان اجرا

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



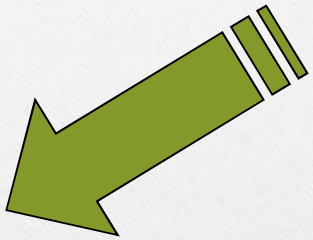
```
#include    <iostream.h>
int main( )
{
cout <<"C++ is an object oriented language \n" ;
return 0 ;
}
```

برنامه زير يك حرف انگليسي كوچك را گرفته به حرف بزرگ تبديل مي‌نمايد.



```
#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; }
```

دو عدد از نوع اعشاري را گرفته مجموع و حاصلضرب آنها را محاسبه و نمايش مي‌دهد.



```
#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 ;
}
```

فصل دوم

ساختارهای تصمیم گیری و تکرار

فهرست مطالب فصل دوم

۱. عملگر های رابطه ای
۲. عملگر شرطی
۳. دستورالعمل شرطی
۴. عملگر کاما
۵. عملگر های منطقی
۶. دستورالعمل For

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:

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

عملگر شرطی

شكل كلي عملگر شرطي بصورت زير مي‌باشد:

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

عملگر شرطي تنها عملگري در C++ مي‌باشد كه داراي سه عملوند مي‌باشد.

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 failed نمایش داده میشود.

دستورالعمل شرطی

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

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

مثال ۱ :

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

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;
}
```

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)

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.

عملگرهای منطقی

با استفاده از عملگرهاي منطقي مي‌توان شرطهاي تركيبي در برنامه ايجاد نمود. عملگرهاي منطقي 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

Or

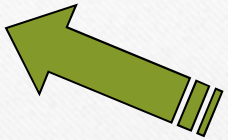
a	b	a || b
true	true	True
true	false	True
false	true	True
false	false	False

Not

a	!a
true	False
false	True

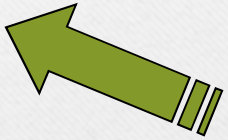
چند مثال :

```
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 .

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" ;
else
cout << "Not a triangle" ;
return  0 ;
}
```

For **دستورالعمل**

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;
 }
```

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 ;
}
```

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 ; }
```

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 ;
}
```

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 ;
}
```

فصل سوم

سایر ساختارهای تکرار

فهرست مطالب فصل سوم

۱. دستورالعمل while
۲. دستورالعمل do while
۳. دستورالعمل break
۴. دستورالعمل continue
۵. دستورالعمل switch
۶. تابع cin.get()
۷. عملگر static_cast<>()
۸. جدول اولویت عملگرها

while دستورالعمل

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 ;
}
```

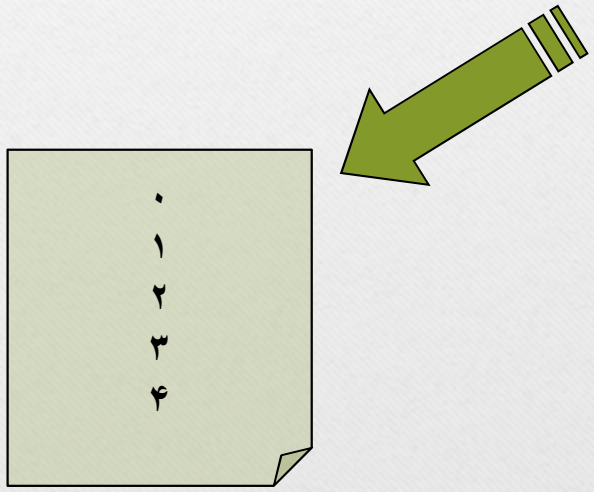
تفاوت دستورهای for و while

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.

مثال :

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

با اجرای قطعه برنامه فوق مقادیر زیر نمایش داده میشود :



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

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

do while **دستورالعمل**

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);
```

تفاوت دستورهای while و do 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 .

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

دستورالعمل break

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

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 ; }
```

مثال ۲:

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

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 ; }
```

continue **دستورالعمل**

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

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 ; }
```

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 ;
 }
```

switch دستورالعمل

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 آنهـا مشـکل میشـود . برای حل این مشکل C++ ، دستورالعمل switch که عملاً یک دستورالعمل چند انتخابی می‌باشد را ارائه نموده است.

switch case

Switch شکل کلی دستور العمل

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

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

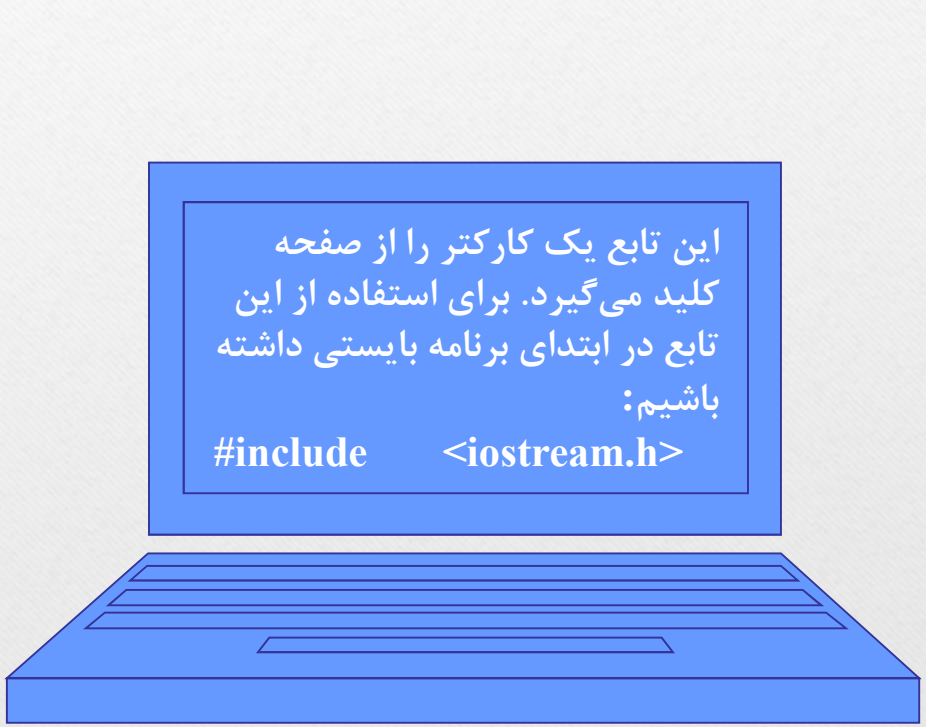
مثال ۱ :

```
#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 * /
}
```

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;
         }
}
```

تابع cin.get() :



قطعه برنامه ذیل یک کرکتر 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;
  }
```

برنامهٔ ذیل یک سطر متن انگلیسی را گرفته کرکترهای 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 ;
 }
```

عملگر static_cast

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

مثال ۱:

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

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

مثال ۲:

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

ابتدا مقدار ۱۴ نمایش داده میشود و سپس مقدار ۱۴,۷۵ نمایش 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	,