تکرار کنترل شده توسط نگهبان

- اگر تعداد دانشجویان کلاس از قبل معلوم نباشد نمی‌توان از برنامه‌ی قبلی برای محاسبه‌ی میانگین و جمع نمرهای کلاس استفاده کرد.
- می‌خواهیم برنامه‌ای بنویسیم که به تعداد دلخواهی نمره از ورودی بخواند و سپس میانگین و جمع نمرهای کلاس را در خروجی چاپ کند.
- چگونه می‌توان به برنامه پیغام داد که نمرها به پایان رسیده‌اند و دیگر نمره‌ای برای وارد کردن وجود ندارد؟
- مقدار نگهبان (مقدار پرچم): مقداری که مشخص کننده‌ی پایان ورود داده‌هاست.
- مقدار نگهبان باید طوری انتخاب شود که با یک مقدار ورودی مجاز اشتباه گرفته نشود.
- به عنوان نمونه، در مثال بعدی، چون نمره‌ها (ورودی‌های مجاز) اعدادی بین صفر تا ۱۰۰ هستند، مقدار نگهبان 1- انتخاب شده است.

مثال ۸

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
   int total = 0, i = 0, grade;
   double average;
   cout << "Enter grade or -1 to quit: ";
   cin >> grade;
   while( grade != -1 )
   {
      total = total + grade;
      i = i + 1;
      cout << "Enter grade or -1 quit: " ;
      cin >> grade;
   }
```

مثال ۸

```
    if( i != 0 )
    {
        average = static_cast<double>(total) / i;
        cout << "\nTotal of all " << i
             << " grades entered is "
             << total << endl;
        cout << "Class average is "
             << setprecision(2) << fixed
             << average << endl;
    }
    else
        cout << "No grades were entered" << endl;

     return 0;
}
```

شرح مثال ۸

- در دستور

```
cout << "Class average is "
     << setprecision(2) << fixed
     << average << endl;
```

- تابع setprecision(2) با آرگومان ۲ احضار شده است.
- این احضار باعث می‌شود متغیر average که از نوع double است با دو رقم دقت در سمت راست نقطه‌ی اعشاری چاپ شود (مثل 92.37) .
- setprecision یک شکل دهنده‌ی پارامتری جریان است که در سرفایل <iomanip> قرار دارد و به فضای نام استاندارد (std) تعلق دارد.
- پس برای استفاده از setprecision باید دستور پیش‌پردازنده‌ی

```
#include <iomanip>
```

در ابتدای برنامه قرار گیرد.

شرح مثال ۸

- اعداد اعشاری می‌تواند به دو صورت نقطه‌ی اعشاری شناور (نماد علمی) یا نقطه‌ی اعشاری ثابت نمایش داده شوند. به عنوان مثال

```
3.82945609E05
382945.609
```

- شکل دهنده‌ی جریان `fixed` در دستور قبل معین می‌کند که مقادیر اعشاری باید در خروجی با قالب (فورمت) نقطه‌ی اعشاری ثابت استفاده شود.
- شکل دهنده‌ی جریان `fixed` به فضای نام استاندارد (`std`) تعلق دارد.
- شکل دهنده‌ی جریان `fixed` مانند `endl` یک شکل دهنده‌ی جریان غیر پارامتری است و در سرفایل `<iostream>` (و نه سرفایل `<iomanip>`) قرار دارد.

عملگرهای جایگزینی

- گاهی نیاز است که به متغیری مقدار مشخصی اضافه یا کم شود یا متغیر در مقدار معینی ضرب یا تقسیم شود سپس مقدار جدید در همان متغیر جایگزین شود. به عنوان مثال

```
i = i + 1;
a = a * 5;
```

- c++ برای کوتاه‌نویسی عبارت‌های جایگزینی، چند عملگر را در اختیار می‌گذارد.
- برای مثال، عبارت

```
z = z + 7;
```

- را می‌توان با کمک عملگر جایگزینی جمع به شکل کوتاه‌تر زیر نوشت

```
z += 7;
```

- عملگر =+ مقدار عملوند سمت راست را با مقدارعملوند سمت راست جمع می‌کند و حاصل را در عملگر سمت چپ جایگزین می‌کند.

عملگرهای جایگزینی

- فهرست عملگرهای جایگزینی

Assignment operator	Sample expression	Explanation	Assigns
Assume: `int c = 3, d = 5, e = 4, f = 6, g = 12;`			
`+=`	`c += 7`	`c = c + 7`	10 to c
`-=`	`d -= 4`	`d = d - 4`	1 to d
`*=`	`e *= 5`	`e = e * 5`	20 to e
`/=`	`f /= 3`	`f = f / 3`	2 to f
`%=`	`g %= 9`	`g = g % 9`	3 to g

عملگرهای افزایشی و کاهشی

- ++C دو عملگر یکانی افزایشی و کاهشی را برای افزایش مقدار یا کاهش مقدار یک متغیر عددی به اندازه‌ی یک واحد در اختیار می‌گذارد.

Operator	Called	Sample expression	Explanation
++	preincrement	++a	Increment a by 1, then use the new value of a in the expression in which a resides.
++	postincrement	a++	Use the current value of a in the expression in which a resides, then increment a by 1.
--	predecrement	--b	Decrement b by 1, then use the new value of b in the expression in which b resides.
--	postdecrement	b--	Use the current value of b in the expression in which b resides, then decrement b by 1.

مثال ۹

```
#include <iostream>
using namespace std;

int main()
{
    int c = 5;
    cout << c << endl;
    cout << c++ << endl;
    cout << c << endl;
    cout << endl;
    c = 5;
    cout << c << endl;
    cout << ++c << endl;
    cout << c << endl;
    return 0;
}
```

شرح مثال ۹

- در دستور

```
cout << c++ << endl;
```

- ابتدا مقدار متغیر c (یعنی 5) در خروجی چاپ و سپس توسط عملگر پس‌افزایش یک واحد به c اضافه می‌شود.

- در دستور

```
cout << ++c << endl;
```

- ابتدا مقدار متغیر c (یعنی 5) توسط عملگر پیش‌افزایش یک واحد اضافه و سپس مقدار جدید c در خروجی چاپ می‌شود.

تمرین

- خروجی حاصل از اجرای برنامه‌ی زیر به چه شکل است؟

```
#include <iostream>
using namespace std;

int main()
{
   int count = 1; // initialize count

   while ( count <= 10 ) // loop 10 times
   {
      // output line of text
      cout << ( count % 2 ? "****" : "++++++++" ) << endl;
      ++count; // increment count
   } // end while
} // end main
```

تمرین

- برنامه‌ی زیر چه مقداری را در خروجی چاپ می‌کند؟

```
#include <iostream>
using namespace std;

int main()
{
   int sum; // stores sum of integers 1 to 10
   int x; // counter

   x = 1; // count from 1
   sum = 0; // initialize sum

   while ( x <= 10 ) // loop 10 times
   {
      sum += x; // add x to sum
      ++x; // increment x
   } // end while

   cout << "The sum is: " << sum << endl;
} // end main
```

تمرین

- خروجی حاصل از اجرای برنامه‌ی زیر به چه شکل است؟

```
#include <iostream>
using namespace std;

int main()
{
   int row = 10; // initialize row
   int column; // declare column

   while ( row >= 1 ) // loop until row < 1
   {
      column = 1; // set column to 1 as iteration begins

      while ( column <= 10 ) // loop 10 times
      {
         cout << ( row % 2 ? "<" : ">" ); // output
         ++column; // increment column
      } // end inner while

      --row; // decrement row
      cout << endl; // begin new output line
   } // end outer while
} // end main
```

تمرین

- برنامه‌ای بنویسید که ۵۰ توان اول عدد ۲ را در خروجی چاپ کند.
- با استفاده از دستور while برنامه‌ای بنویسید که ۱۰۰ عدد اعشاری را از ورودی بخواند، بزرگترین عدد در بین عددهای وارد شده را با پیغام مناسب در خروجی چاپ کند.
- برنامه‌ای بنویسید که یک عدد صحیح را از ورودی دریافت کند، فاکتوریل آن را محاسبه و در خروجی چاپ کند.
- برنامه‌ای بنویسید که مقدار تقریبی ثابت ریاضی e را با استفاده از بسط

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots = \sum_{i=0}^{\infty} \frac{1}{i!}$$

محاسبه کند. برای تعیین میزان دقت تقریب، برنامه باید با چاپ پیغام مناسب از کاربر بخواهد تعداد جملاتی که برای تقریب باید جمع شوند را وارد کند.