

## دستور تکرار (حلقه) while

- صورت کلی استفاده از دستور while به شکل زیر است

`while` ( شرط حلقه )

{

بدنه‌ی دستور `while`

}

- تا وقتی که شرط حلقه درست است دستورهای بدنه‌ی حلقه به طور مکرر اجرا می‌شوند.
- اگر بدنه‌ی دستور `while` تنها شامل یک دستور باشد، استفاده از گروه برای تعیین بدنه‌ی دستور `while` ضروری نیست.
- اگر شرط حلقه در ابتدا نادرست باشد دستورهای حلقه هرگز اجرا نمی‌شوند.
- اگر شرط حلقه همواره درست باشد، حلقه بی‌نهایت بار تکرار شده و اجرای برنامه هرگز به پایان نمی‌رسد مگر آن که عاملی خارج از برنامه (کاربر یا سیستم عامل) اجرای برنامه را متوقف کند.



## مثال ۶: مضرب‌های عدد ۳

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

int main()
{
    int m = 3;
    while (m <= 100)
        m = m * 3;

    cout << m << endl;
    return 0;
}
```



## شرح مثال ۶

- این برنامه اولین عدد توان ۳ که بزرگتر از ۱۰۰ می‌باشد را پیدا می‌کند و در خروجی چاپ می‌کند.
- در آغاز مقدار متغیر  $m$  برابر ۳ است. توجه کنید که مقدار دهی اولیه را می‌توان در اعلان متغیر انجام داد.
- با هر بار تکرار بدنه‌ی دستور `while`، مقدار متغیر  $m$  در ۳ ضرب می‌شود و مقدار جدید (۹، ۲۷، ۸۱، ۲۴۳ و ...) در متغیر  $m$  جایگزین می‌شود.
- وقتی مقدار  $m$  برابر ۲۴۳ شد، شرط حلقه نادرست است و لذا تکرار حلقه به پایان می‌رسد.
- بنابراین مقدار نهایی  $m$  عدد ۲۴۳ است که در خروجی چاپ می‌شود.



## مثال ۷: تکرار کنترل شده با شمارنده

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

int main()
{
    int total, i, grade;
    double average;
    total = 0;
    i = 1;
    while (i <= 10)
    {
        cout << "Enter grade: ";
        cin >> grade;
        total = total + grade;
        i = i + 1;
    }
}
```



## مثال ۷: تکرار کنترل شده با شمارنده

```
average = total / 10;
cout << "\nTotal of all grades is "
      << total << endl;
cout << "Class average is "
      << average << endl;

return 0;
}
```



## شرح مثال ۷

- از متغیر total برای ذخیره‌ی مجموع نمره‌ها استفاده می‌شود و بنابراین به آن مقدار اولیه صفر داده می‌شود.
- متغیر i یک شمارنده برای کنترل تعداد دفعات تکرار حلقه (تعداد دفعات خواندن نمره‌ها) است. به همین دلیل به آن مقدار اولیه‌ی ۱ داده شده است.
- از متغیر average برای ذخیره‌ی میانگین کلاس استفاده می‌شود.
- متغیر average از نوع double اعلان شده است.
- ++C برای ذخیره‌ی اعداد اعشاری از چند نوع داده‌های مختلف مانند float و double استفاده می‌کند.
- متغیرهای از نوع float اعداد اعشاری را با دقت یگانه ذخیره می‌کنند: در کامپیوترهای ۳۲ بیتی امروزی چنین اعدادی دارای ۷ رقم معنی‌دار هستند.
- متغیرهای از نوع double اعداد اعشاری را با دقت دوگانه (مضاعف) ذخیره می‌کنند. پس برای متغیرهای double بیشتر از متغیرهای float حافظه نیاز است.



## شرح مثال ۷

- کامپیوترهای ۳۲ بیتی امروزی اعداد اعشاری را تا ۱۵ رقم معنی دار ارائه می کنند.
- با توجه به بالا بودن حافظه‌ی اصلی کامپیوترهای شخصی امروزی در ادامه از نوع `double` برای ذخیره‌سازی اعداد اعشاری استفاده می کنیم.
- در دستور

```
average = total / 10;
```

- مقدار متغیر `total` و ۱۰ هر دو عدد صحیح هستند و بنابراین حاصل تقسیم نیز به شکل صحیح خواهد بود یعنی قسمت اعشاری نتیجه‌ی تقسیم بریده می شود.
- پس از مقدار `total` برابر ۸۴۶ باشد، نتیجه‌ی تقسیم `total / 10` به جای 84.6، 84 است.
- با استفاده از عملگر یکانی تبدیل نوع

```
static_cast<double>(total)
```

- یک کپی اعشاری از عملوند (مقدار متغیر `total`) ایجاد می شود.



## شرح مثال ۷

- پس در دستور  

```
average = static_cast<double>(total) / 10;
```
- یک عدد اعشاری (نسخه‌ی total موقت از total) را بر عدد صحیح ۱۰ تقسیم می‌شود.
- برای ارزیابی عبارتی که در آن نوع داده‌ای عملوندها یکسان نیستند عمل ارتقاء نوع (تبدیل نوع ضمنی) صورت می‌گیرد.
- به عنوان مثال در عبارت‌هایی که حاوی مقادیری از نوع int و double هستند، عملوندهای int به مقادیر double ارتقاء داده می‌شوند.
- پس در دستور بالا عدد صحیح ۱۰ به نوع double ارتقاء داده می‌شود و نتیجه‌ی تقسیم اعشاری در متغیر average ذخیره می‌شود.
- آیا می‌توانستیم در عوض از دستور زیر استفاده کنیم؟  

```
average = total / 10.0;
```

