

# تکرار با شمارنده

- تکرار با شمارنده یکی از متداول‌ترین انواع تکرار است.
- برای تکرار با شمارنده باید چند چیز را در نظر داشت:
  1. یک متغیر به عنوان شمارنده
  2. مقداردهی اولیه به شمارنده
  3. شرط حلقه که مشخص می‌کند مقدار پایانی شمارنده چقدر است
  4. نمو شمارنده که مشخص می‌کند مقدار شمارنده در هر تکرار باید چگونه تغییر کند



## مثال ۱۰ : چاپ اعداد ۱ تا ۱۰

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

int main()
{
    int counter = 1;

    while (counter <= 10)
    {
        cout << counter << " ";
        counter++;
    }

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



# دستور تکرار for

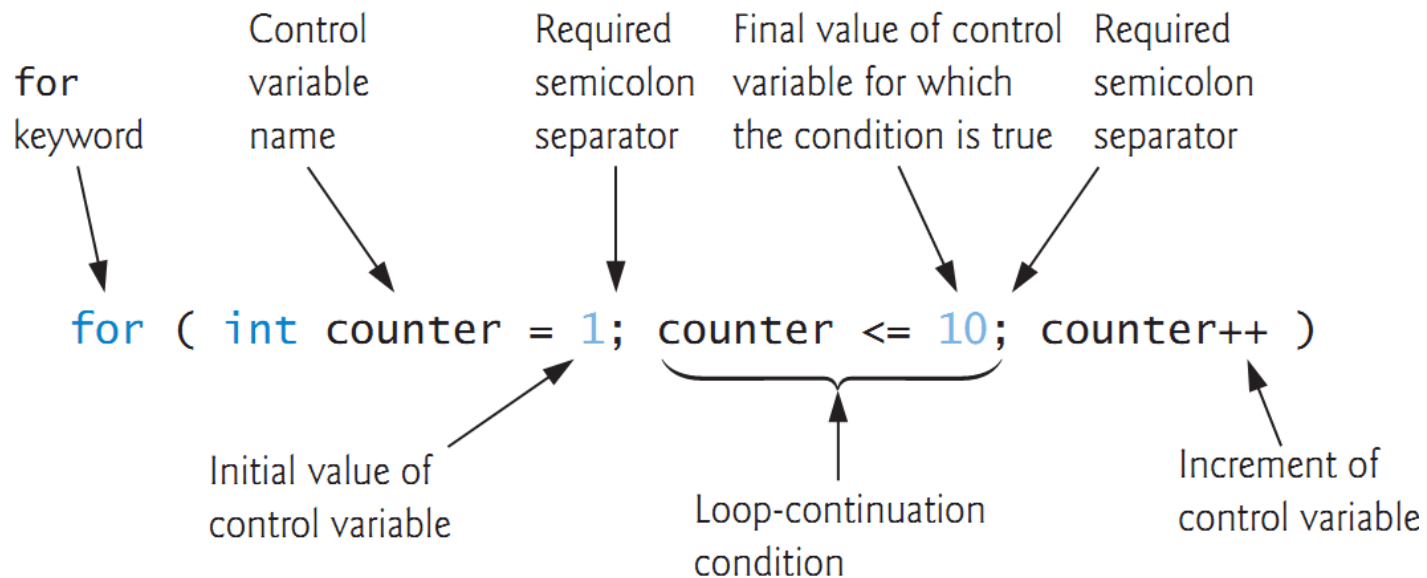
- دستور for جزئیات تکرار کنترل شده توسط شمارنده را در یک خط از برنامه مشخص می کند.

(نمو شمارنده ; مقدار پایانی شمارنده ; شمارنده و مقدار اولیه) for

```
{
```

بدنه ی دستور for

```
}
```



## مثال ۱۱: چاپ اعداد ۱ تا ۱۰

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

int main()
{
    for (int counter=1; counter<=10; counter++)
        cout << counter << " ";

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



## مثال

```
for (int i = 1; i <= 100; i++)
```

- متغیر کنترلی را از ۱ تا ۱۰۰ در هر تکرار به اندازه‌ی یک واحد افزایش می‌دهد.

```
for (int i = 100; i >= 1; i--)
```

- متغیر کنترلی را از ۱۰۰ تا ۱ در هر تکرار به اندازه‌ی یک واحد کاهش می‌دهد.

```
for (int i = 7; i <= 77; i += 7)
```

- متغیر کنترلی را از ۷ تا ۷۷ در هر تکرار به اندازه‌ی هفت واحد افزایش می‌دهد.

```
for (int i = 99; i >= 0; i -= 11)
```

- متغیر کنترلی را از ۹۹ تا صفر در هر تکرار به اندازه‌ی ۱۱ واحد کاهش می‌دهد.



## مثال ۱۲: جمع اعداد زوج کمتر مساوی ۲۰۰

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

int main()
{
    int total = 0;
    for (int numb = 2; numb <= 200; numb += 2)
        total += numb;

    cout << "sum is " << total << endl;

    return 0;
}
```



## دستور تکرار for

- متغیر کنترلی حلقه‌ی for را می‌توان خارج از دستور و در ابتدای برنامه نیز اعلان کرد.
- اگر متغیر کنترلی حلقه‌ی for در خود دستور اعلان شده باشد، تنها می‌توان از آن درون حلقه استفاده کرد اما نمی‌توان از آن در خارج از دستور for استفاده کرد.
- این استفاده‌ی محدودی از نام یک متغیر حوزه‌ی فعالیت آن متغیر نام دارد.
- حوزه‌ی فعالیت یک متغیر عبارتست از مکانی از برنامه که متغیر می‌تواند در آن مورد استفاده قرار گیرد.
- در مثال قبل، حوزه‌ی فعالیت متغیر total بعد از اعلان همه‌ی بدنه‌ی تابع main است.
- حوزه‌ی فعالیت متغیر numb تنها درون بدنه‌ی حلقه‌ی for است.
- سه عبارت درون دستور for اختیاری هستند.
- وجود دو سمی‌کالن در دستور for اجباری است.
- مثال (حلقه‌ی بی‌پایان)

```
for (int numb = 2; ; numb += 2)
```



## مثال ۱۳

- با فرض این که نرخ سود سالانه‌ی حساب سپرده‌گذاری یک بانک ۵٪ باشد، برنامه‌ای بنویسید که با سپرده‌گذاری ۱۰۰۰ دلاری، مقدار سرمایه‌ی سالانه تا ۱۰ سال را در خروجی چاپ کند.

$$a = p(1 + r)^n$$

- سرمایه اولیه:  $p$
- نرخ سود سالانه:  $r$
- مدت زمان سپرده‌گذاری (بر حسب سال):  $n$
- مقدار سرمایه‌ی در پایان دوره سپرده‌گذاری:  $a$





## مثال ۱۳

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

int main()
{
    double amount, principal=1000.0, rate=0.05;

    cout << "Year" << setw(21)
         << "Amount on deposit" << endl;

    cout << fixed << setprecision(2);
```



## مثال ۱۳

```
for (int year = 1; year <= 10; year++)
{
    amount = principal *
                pow(1.0 + rate, year);
    cout << setw(4) << year << setw(21)
        << amount << endl;
}

return 0;
}
```



## شرح مثال ۱۳

- تابع pow برای محاسبه‌ی توان به کار می‌رود و در سرفایل تابع‌های ریاضی cmath قرار دارد.
- شکل‌دهنده‌ی جریان setw یک شکل‌دهنده‌ی جریان پارامتری متعلق به فضای نام استاندارد (std) است که برای استفاده از آن باید سرفایل iomanip در برنامه گنجانده شود.
- شکل‌دهنده‌ی جریان setw با آرگومان 4 مشخص می‌کند که مقدار بعدی خروجی باید در میدانی به طول چهار کاراکتر چاپ شود؛ یعنی cout خروجی بعدی را در فضایی با دست کم چهار کاراکتر چاپ می‌کند.
- خروجی برنامه:

Year	Amount on deposit
1	1050.00
2	1102.50
3	1157.63
4	1215.51
5	1276.28
6	1340.10
7	1407.10
8	1477.46
9	1551.33
10	1628.89



## شرح مثال ۱۳

- در اعمال حسابی بر روی داده‌های مالی (مقدرهای پولی) باید یک نکته را مد نظر داشت.
- واحدهای پولی کشورهای مختلف از یک واحد اصلی و یک واحد کسری (یک یا دو رقم اعشار) تشکیل شده است: دلار واحد اصلی و سنت واحد کسری (تا دو رقم)
- بنابراین اعداد با رقم اعشار بیش از تعداد ارقام واحد کسری فاقد معنی هستند و در عمل گرد می‌شوند.
- به عنوان مثال حاصل یک سری محاسبات بر روی داده‌های پولی ممکن است مقدار 14.234 را بدست دهد که با گرد کردن تا دو رقم اعشار مقدار 14.23 در خروجی چاپ می‌شود. همین‌طور برای 18.673 در خروجی مقدار 18.67 مشاهده می‌شود. اگر این دو عدد با هم جمع شوند مقدار 32.907 بدست می‌آید که در هنگام چاپ با دقت دو رقم اعشار در خروجی 32.91 درج می‌شود در حالی که  $14.23+18.67=32.90$

