

مبانی کامپیوتر و برنامه سازی

دانشگاه آل طه

رشته کامپیوتر

درس تخصصی

پاییز 1403

رئوس مطالب

- | | |
|--|---|
| ☐ آرایه ها | ☐ مفاهیم اولیه |
| ☐ اشاره گرها | ☐ محاسبات در کامپیوتر |
| ☐ کاراکترها و رشته ها | ☐ آشنایی با الگوریتم و فلوچارت |
| ☐ ساختارها | ☐ مقدمات برنامه سازی |
| ☐ آشنایی با ++c | ☐ فرمت بندی ورودی/خروجی |
| ☐ آشنایی با رده ها | ☐ دستورها |
| ☐ حوزه های public , private | ☐ توابع |
| ☐ پرونده ها | ☐ آزمون و خطایابی برنامه |

ساختار حلقه for در زبان برنامه نویسی C++

```
for(;;){
```

```
    دستورات
```

```
}
```

```
for(;;)
```

```
    دستور
```

برای ایجاد حلقه تکرار بی نهایت (شرط پایان ندارد)

راه خروج: `ctrl+break`

ساختار حلقه for در زبان برنامه نویسی C++

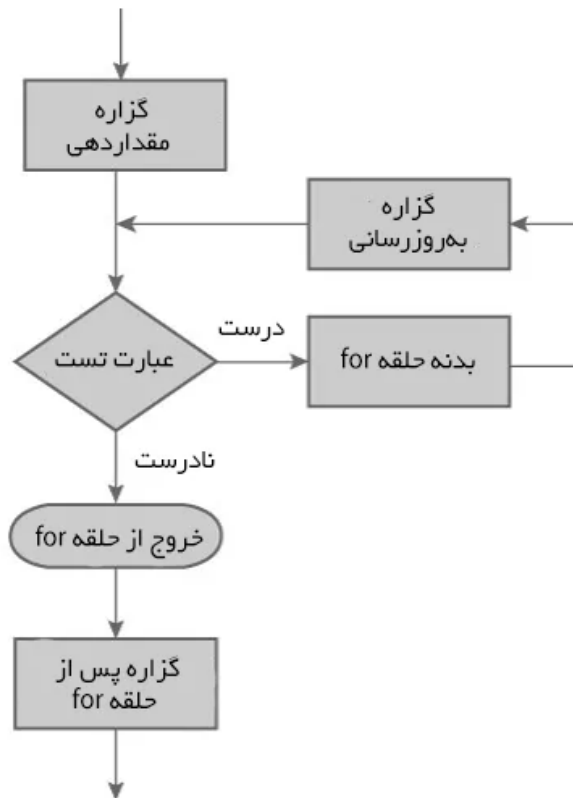
```
{(گام حرکت; شرط حلقه; مقدار اولیه اندیس حلقه)for  
دستورات;  
}
```

```
-----  
(گام حرکت; شرط حلقه; مقدار اولیه اندیس حلقه)for  
دستور;
```

طرز کار حلقه for

- گزاره مقداره‌ی تنها یک بار و در ابتدای کار اجرا می‌شود.
- سپس عبارت تست ارزیابی می‌شود.
- اگر عبارت تست نادرست باشد، حلقه for خاتمه می‌یابد؛ اما اگر عبارت تست true باشد، کدهای درون بدنه حلقه for اجرا می‌شوند و عبارت update به‌روزرسانی می‌شود.
- یک بار دیگر عبارت تست ارزیابی می‌شود و این فرایند تا زمانی که عبارت تست false شود، تکرار می‌شود.

فلوچارت حلقه for در C++



حلقه for تو در تو در C++

```
for(int j=1;j<=m;j++)  
    {  
        .....  
        for(int i=1;i<=n;i++)  
            {  
                .....  
            }  
        .....  
    }
```

گزاره if در C++

```
if(شرط){  
    دستورات;  
}  
else {  
    دستورات;  
}
```

```
if(شرط)  
    دستور;  
else  
    دستور;
```


توضیح

- گزاره if به ارزیابی عبارت تست درون پرانتز می پردازد.
- اگر عبارت تست، درست ارزیابی شود، گزاره های درون بدنه if اجرا می شوند.
- اگر عبارت تست نادرست ارزیابی شود، گزاره های درون بدنه if رد می شوند.

عبارت تست درست است.

```
int test = 5;
```

```
if (test < 10)
```

```
{
```

```
→ // codes
```

```
}
```

```
// codes after if
```

عبارت تست نادرست است

```
int test = 5;
```

```
if (test > 10)
```

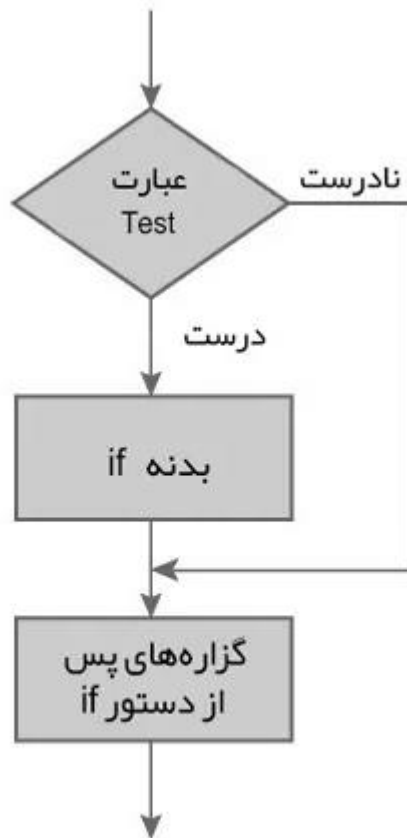
```
{
```

```
→ // codes
```

```
}
```

```
→ // codes after if
```

فلوچارت گزاره if



طرز کار گزاره if...else چگونه است؟

عبارت تست درست است.

```
int test = 5;

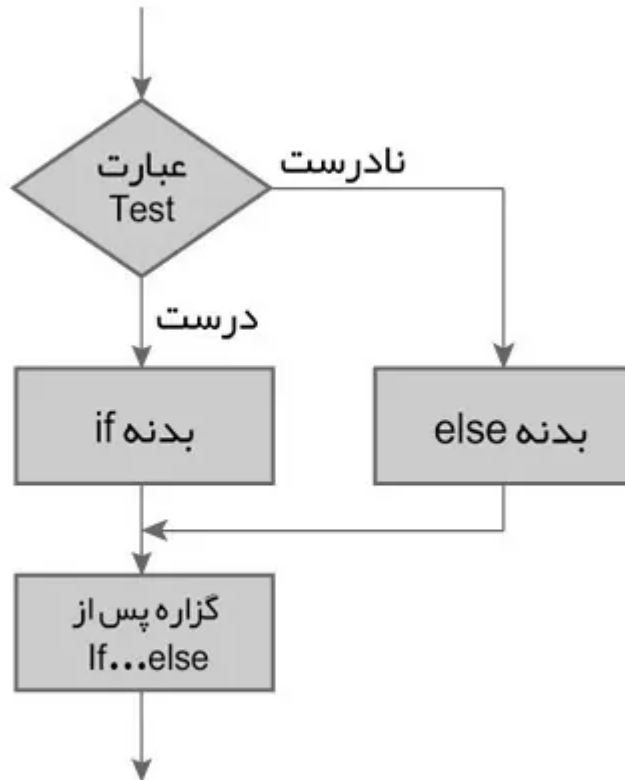
if (test < 10)
{
    // codes
}
else
{
    // codes
}
// codes after if...else
```

عبارت تست نادرست است.

```
int test = 5;

if (test > 10)
{
    // codes
}
else
{
    // codes
}
// codes after if...else
```

فلوچارت گزاره if...else



گزاره `if...else` و `if` تو در تو در `C++`

- گزاره `if...else` وابسته به این که عبارت تست درست یا نادرست باشد، دو کد متفاوت را اجرا می کند. برخی اوقات باید از میان بیش از دو گزینه انتخاب کنیم.
- گزاره `if...else` تو در تو امکان بررسی چندین عبارت تست و اجرای کدهای مختلف برای بیش از دو شرایط متفاوت را در اختیار ما قرار می دهد.

گزاره if...else...else if در C++

```
if (شرط)
{
    دستورات;
}
else if (شرط)
{
    دستورات;
}
else
{
    دستورات;
}
```

عملگر شرطی/سه‌تایی (?)

```
1  if ( a < b ) {  
2      a = b;  
3  }  
4  else {  
5      a = -b;  
6  }
```

```
1  a = (a < b)? b: -b;
```

حلقه while در C++

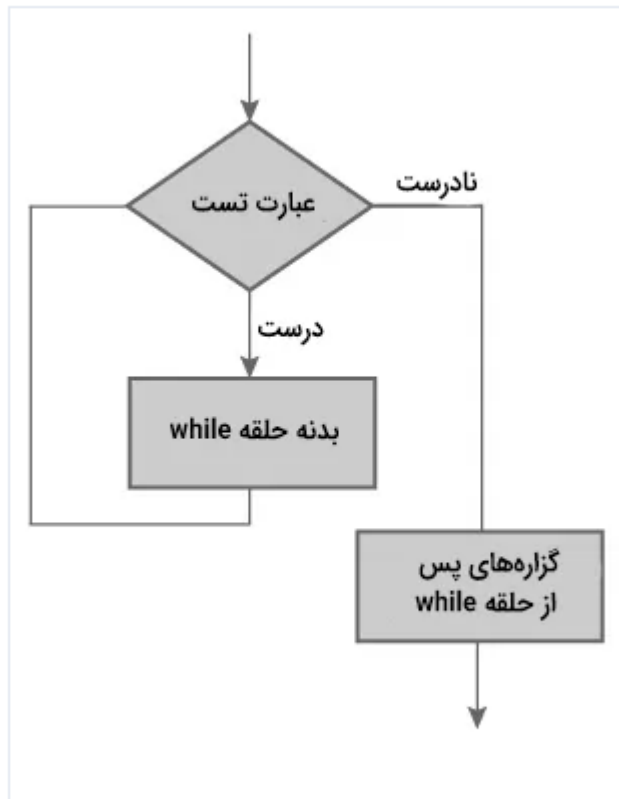
```
while (شرط)
{
    دستورات;
}
```

```
while (شرط)
    دستور;
```


حلقه while چگونه کار می کند؟

- حلقه while عبارت تست را ارزیابی می کند.
- اگر عبارت تست درست باشد، کد درون بدنه حلقه while مورد ارزیابی قرار می گیرد.
- سپس عبارت تست مجدداً ارزیابی می شود. این فرایند تا زمانی که عبارت تست نادرست شود ادامه می یابد.
- هنگامی که عبارت تست نادرست شود، حلقه while خاتمه می یابد.

فلوچارت حلقه while



حلقه do...while در C++

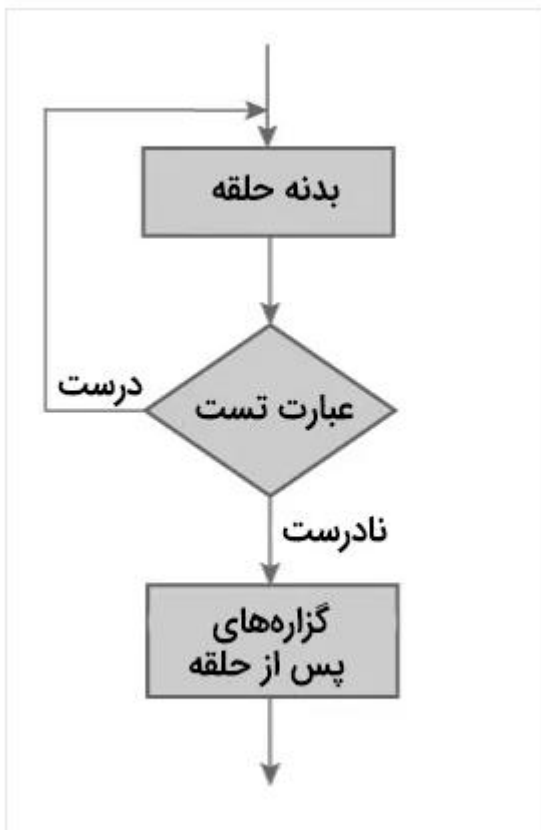
```
do {  
    دستورات;  
}  
while (شرط);
```

```
-----  
do  
    دستور;  
while (شرط);
```

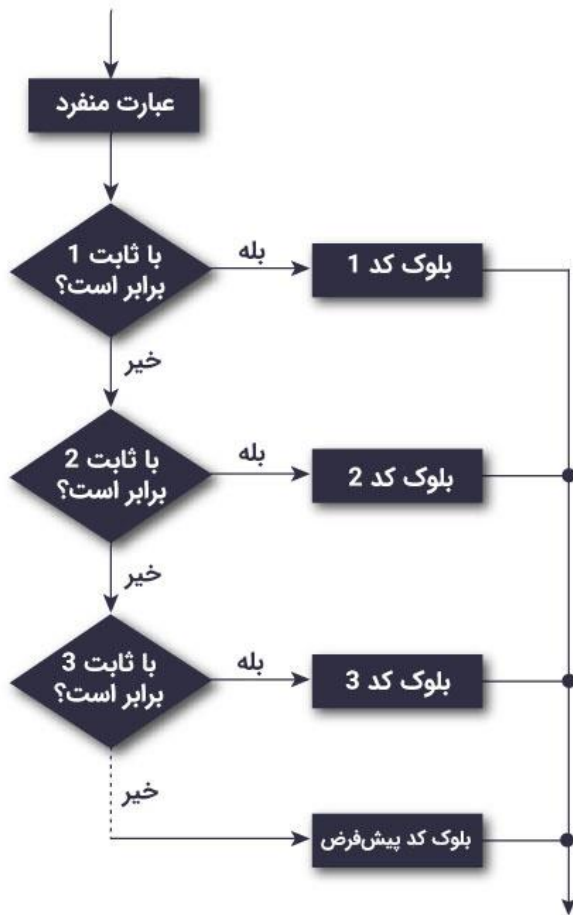
طرز کار حلقه do...while چگونه است؟

- کدهای درون بدنه حلقه دست کم یک بار اجرا می شوند.
سپس تنها عبارت تست بررسی می شود.
- اگر عبارت تست درست باشد، بدنه حلقه اجرا می شود. این فرایند تا زمانی ادامه می یابد که عبارت نادرست شود.
- هنگامی که عبارت تست نادرست باشد، حلقه do...while خاتمه می یابد.

فلوچارت حلقه do...while



ساختار تصمیم‌گیری switch در C++



```
switch (expression) {  
    case constant1:  
        // دستورات مربوط به case 1  
        break;  
    case constant2:  
        // دستورات مربوط به case 2  
        break;  
    // ...  
    default:  
        // دستورات مربوط به حالت پیش‌فرض (اختیاری)  
}  
}
```

ساختار تصمیم‌گیری switch در ++C

مقایسه یک مقدار با چندین مقدار ثابت

وقتی یک ثابت حالت پیدا شود که با عبارت سوئیچ مطابقت پیدا کند، کنترل برنامه به بلوک کدی انتقال می‌یابد که با آن حالت مرتبط است.

گزاره break برای جلوگیری از اجرای کد در case بعدی استفاده می‌شود.