

ساختارهای تصمیم در برنامه نویسی C++

در برنامه هرگاه بخواهیم در شرایط ویژه ای برخی از دستورات اجرا شوند و برخی دیگر اجرا نشوند از ساختارهای تصمیم استفاده می کنیم.

• ساختار: if

با استفاده از ساختار **if** ، شرطی را کنترل می کنیم. اگر شرط جلوی **if** درست باشد دستورات آن اجرا خواهد شد، در غیر اینصورت دستورات درون **else** اجرا خواهند شد.

توجه کنید که فقط و فقط یا دستورات **if** اجرا می شود و یا دستورات **else** و هرگز این دو با هم اجرا نمی شوند.

if (عبارت شرطی)

```
{  
    Line Command 1;  
    Line Command 2;  
    Line Command n;  
}
```

در ادامه به بررسی حالات مختلفی در بکارگیری دستور **if** در C++ می پردازیم:

```
۱ int num = 5;  
۲ if ( num == 10 )  
۳     cout << "num is equal 10" ;  
۴
```

در کد برنامه نویسی بالا می بینیم که **num** برابر ۵ است، بنابراین شرط **if** نادرست است و دستور مربوط به آن چاپ نمی شود.

```
۱ int num = 10;  
۲ if ( num == 10 )  
۳ {  
۴     cout << "num is :" ;  
۵     cout << num ;  
۶ }
```

۶
۷

```
num is :10
```

در کد برنامه نویسی بالا چون دستورات مربوط به `if` از یک سطر بیشتر است آنها را در بلوک آکولاد قرار دادیم. می بینیم که این بار مقدار `num` برابر با شرط است، در نتیجه دستورات مربوط به `if` اجرا می شوند.

نکته مهم اینجاست که برای بررسی تساوی از `==` استفاده کردیم و دلیل آن این است که این عملگر یک عملگر رابطه ای است و در بررسی شروط باید از عملگرهای رابطه ای استفاده کنیم.

```
1 int num = 5;
2 if ( num == 10 )
3 {
4     cout << "num is :";
5     cout << num ;
6 }
7 else
8 {
9     cout << "num is not 10 % is :";
10    cout << num ;
11 }
12
```

```
num is not 10 and is :5
```

در کد برنامه نویسی بالا چون `num` برابر با ۱۰ نیست پس دستورات درون `else` اجرا می شود.

مطالب قبلی دربرگیرنده ۲ حالت از بررسی شرط `if` بودند، اما گاهی اوقات لازم است که ما حالات بیشتری را بررسی نماییم که در اینصورت از `else if` بهره می گیریم.

```
۱ if ( num > 0 )  
۲     cout << "num is positive ";  
۳ else if ( num < 0 )  
۴     cout << "num is negative ";  
۵ else  
۶     cout << "num is 0 ";
```