

عملگرهای برابری و مقایسه‌ای

- عملگرهای برابری و مقایسه‌ای (تصمیم‌گیری) در C++ به قرار زیر هستند:
 - عملگرهای برابری (اولویت یکسانی دارند)

| معنی | مثال | C++ | عملگر |
|-------------------|----------|-----------------|----------------|
| X برابر با Y است | $x == y$ | <code>==</code> | <code>=</code> |
| X برابر با Y نیست | $x != y$ | <code>!=</code> | <code>≠</code> |

- عملگرهای مقایسه‌ای (اولویت یکسان دارند اما اولویت آن‌ها از عملگرهای برابری بالاتر است)

| معنی | مثال | C++ | عملگر |
|-------------------------|------------|--------------------|-------------------|
| X بزرگتر از Y است | $x > y$ | <code>></code> | <code>></code> |
| X کوچکتر از Y است | $x < y$ | <code><</code> | <code><</code> |
| X بزرگتر یا مساوی Y است | $x \geq y$ | <code>>=</code> | <code>≥</code> |
| X کوچکتر یا مساوی Y است | $x \leq y$ | <code><=</code> | <code>≤</code> |

- نتیجه (مقدار) عملگرهای مقایسه‌ای و برابری یا درست (true) یا نادرست (false) است و به همین خاطر از آن‌ها به عنوان شرط استفاده می‌شود.



مثال ۳: استفاده از دستور if

```
// Comparing integers using if statements
#include <iostream>
using namespace std;

int main()
{
    int number1, number2;
    cout << "Enter two integers to compare: ";
    cin >> number1 >> number2;

    if (number1 == number2)
        cout << number1 << "==" << number2 << endl;
    if (number1 != number2)
        cout << number1 << "!=" << number2 << endl;
    if (number1 < number2)
        cout << number1 << "<" << number2 << endl;
}
```



مثال ۳: استفاده از دستور if

```
if (number1 > number2)
    cout << number1 << ">" << number2 << endl;
if (number1 <= number2)
    cout << number1 << "<=" << number2 << endl;
if (number1 >= number2)
    cout << number1 << ">=" << number2 << endl;

return 0;
}
```



شرح مثال ۳

- با استفاده از دستور زیر نیازی به قبل از استفاده از `cout`، `cin` و `endl` نیست.

```
using namespace std;
```

- دستور زیر دو عدد صحیح را از جریان ورودی استاندارد می‌خواند و در دو متغیر `number1` و `number2` ذخیره می‌کند.

```
cin >> number1 >> number2;
```

- همانند جریان خروجی، در جریان ورودی نیز می‌توان از چند عملگر استخراج از جریان استفاده کرد.

- صورت کلی استفاده از دستور `if` به شکل زیر است

```
if ( شرط )
```

```
{
```

```
    بدنه‌ی دستور if
```

```
}
```

- اگر شرط دستور `if` برقرار باشد؛ یعنی ارزش شرط درست (`true`) باشد، آن‌گاه دستورهای بدنه‌ی دستور `if` به ترتیب اجرا می‌شوند.



شرح مثال ۳

- اگر شرط برقرار نباشد؛ یعنی ارزش شرط نادرست (false) باشد، دستورهای بدنه‌ی دستور if اجرا نمی‌شوند.
- اگر بدنه‌ی دستور if تنها شامل یک دستور باشد، استفاده از کروشه برای تعیین بدنه‌ی دستور if ضروری نیست.
- قرار دادن سمی کالن بلافاصله بعد از پرانتز بسته‌ی شرط دستور if اغلب منجر به خطای منطقی می‌شود اما خطای دستوری نیست.

```
if ( شرط ) ;
```

```
{
```

```
    بدنه‌ی دستور if
```

```
}
```

- این سمی کالن باعث خالی در نظر گرفته شدن دستور if می‌شود و بنابراین دستور if صرف نظر از درست یا نادرست بودن شرط آن، هیچ کاری انجام نمی‌دهد. بدتر از آن، دستورهای بدنه‌ی دستور if تبدیل به دستورهای بعد از if می‌شوند و لذا همیشه اجرا خواهند شد.



تمرین

- برنامه‌ای بنویسید که یک عدد پنج رقمی را از ورودی بخواند، آن‌گاه هریک از رقم‌های این عدد (یکان، دهگان، صدگان، هزارگان و ده هزارگان) را به دست آورد، و آن‌ها را با سه فاصله‌ی خالی از یکدیگر در خروجی چاپ کند.
- برنامه‌ای به زبان C++ بنویسید که سه عدد صحیح را از صفحه کلید دریافت کند، آن‌گاه میانگین، کوچکترین مقدار و بزرگترین مقدار این عددها را در خروجی چاپ کند.

