آموزش C++ - درس ۶: پردازش شرطی- قسمت دوم

دستورات switch و عملگرهای منطقی

به سری آموزشهای C++ سایت About خوش آمدید. این دومین درس در مورد پردازش شرطی است که دستور switch و عملگرهای منطقی را معرفی می کند. دستور switch ساختاری است که به جای دستورات if/else بسیار تو در تو یا زنجیره ای استفاده می شود. دستورات if/else زنجیره ای هنگامی رخ می دهد که چندین انتخاب در بعضی از شرطها وجود دارد. در اینجا مثالی می آوریم. فرض کنید که یک بستنی فروش از ما می خواهد که برنامه ای برایش بنویسیم که سفارشهایش را بصورت خودکار انجام دهد. ما نیاز به ارائه یک منو خواهیم داشت و سپس بر مبنای انتخاب کاربر عمل مناسبی را در نظر می گیریم.

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

int main()
{
   int choice;

   cout << "What flavor ice cream do want?" << endl;
   cout << "Enter 1 for chocolate" << endl;
   cout << "Enter 2 for vanilla" << endl;
   cout << "Enter 3 for strawberry" << endl;
   cout << "Enter 4 for green tea flavor, yuck" << endl;
   cout << "Enter you choice: ";

   cin >> choice;

  if (choice == 1) {
      cout << "Chocolate, good choice" << endl;
   }
   else if (choice == 2) {
      cout << "Vanillarific" << endl;
   }
   else if (choice == 3) {
      cout << "Berry Good" << endl;
   }
   else if (choice == 4) {
      cout << "Big Mistake" << endl;
   }
   else {
     cout << "We don't have any" << endl;
      cout << "Make another selection" << endl;
   }

   return 0;
}
```

این برنامه بدرستی کار خواهد کرد، اما نوشتن بلاک if/else طاقت فرسا است. ما می خواهیم آن آسان باشد مخصوصا اگر در اینجا انتخابهای بیشتری وجود داشت و ممکن است انتخابهای تو در تو if/else های بیشتری را برای ساخت برنامه درگیر کند. در اینجا همان مثال با استفاده از یک switch نشان داده شده است.

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

int main()
{
   int choice;

   cout << "What flavor ice cream do want?" << endl;
   cout << "Enter 1 for chocolate" << endl;
   cout << "Enter 2 for vanilla" << endl;
   cout << "Enter 3 for strawberry" << endl;
   cout << "Enter 4 for green tea flavor, yuck" << endl;
   cout << "Enter you choice: ";

   cin >> choice;

   switch (choice) {
```

```
    case 1:
        cout << "Chocolate, good choice" << endl;
        break;
    case 2:
        cout << "Vanillarific" << endl;
        break;
    case 3:
        cout << "Berry Good" << endl;
        break;
    case 4:
        cout << "Big Mistake" << endl;
        break;
    default:
        cout << "We don't have any" << endl;
        cout << "Make another selection" << endl;
    }

    return 0;
}
```

شکل کلی یک دستور switch بدین شکل است:

```
switch (variable)
{
    case expression1:
        do something 1;
        break;
    case expression2:
        do something 2;
        break;
      ....
    default:
        do default processing;
}
```

هر عبارت (expression) باید یک مقدار ثابت باشد. مقدار Variable برای بررسی تساوی با هر عبارت مقایسه می شود. استفاده از عملگرهای رابطه ای دیگر که در درسهای گذشته شرح داده شد یا عملگرهای منطقی که در ادامه همین درس بررسی خواهند شد، امکان پذیر نیست. وقتی یک عبارت پیدا شد که مساوی متغیر بود، دستورات مربوط به آن عبارت تا زمانی که با دستور break مواجه شوند ادامه پیدا می کنند. داشتن یک case بدون دستور break نیز امکان پذیر است. این کار موجب می شود تا عملیات اجرا در درون case دیگر نیز ادامه پیدا کند. این امکان در بعضی مواقع بسیار مفید است. فرض کنید ما باید بر این اساس تصمیم بگیریم که حرف ذخیره شده در متغیر یک حرف صدادار است یا بی صدا.

```
switch (myLetter) {
    case 'A':
    case 'E':
    case 'I':
    case 'O':
    case 'U':
        vowelCnt++; // increments vowel count
            // same as, vowelCnt = vowelCnt + 1;
        break;
    default:
        consonantCnt = consonantCnt + 1;
}
```

اگر هرحرف صدادار در هر کدام از case ها یافت شود مقدار آن یکی افزایش داده شده و دستور break اجرا می شود. برای تمرین سعی کنید که همین برنامه را با if/else بنویسید.

در مثال واقعی تر ممکن است لازم باشد چندین شرط برای تعیین قسمتهایی از کد که باید اجرا شوند، معین کنیم. برای مثال اگر شرط ۱ درست (true) باشد و شرط ۲ هم درست (true) باشد یک پردازش انتخاب شود، اگر شرط ۱ درست (true) باشد و شرط ۲ نادرست (false) باشد پردازش دیگری انتخاب شود. C++ بدین منظور چندین عملگر منطقی را فراهم کرده که اجازه داشتن عبارتهای رابطه ای پیچیده تر و ارزیابی آنها را می دهد.

عملگر	توضیح	مثال	ارزیابی
&&	AND	(5 > 3) AND (5 > 6)	FALSE
&&	AND	(5 > 3) AND (5 > 4)	TRUE
||	OR	(5 > 3) OR (5 > 6)	TRUE
||	OR	(5 > 3) OR (5 > 4)	TRUE
!	NOT	!(5 > 3)	FALSE
!	NOT	!(5 > 6)	TRUE

جدول عملگرهای منطقی در C++

همانطوری که می توانید در این جدول هم بینید، && فقط در صورتی مقدار درست (true) بر می گرداند که هر دو عبارت درست باشند، در حالی که || فقط در صورتی که هر کدام از عبارتها درست (true) باشند مقدار درست (true) بر می گرداند. عملگر ! یک عمل نفی منطقی را فراهم می کند. یکی از موارد خیلی مهم وقتی است که عبارتهای ساخته شده بصورت لیستی از تقدم عملگرهای منطقی و رابطه ای باشد. عملگرهای رابطه ای دارای تقدم بالاتری از عملگرهای منطقی هستند و تقدم عملگرها از چپ به راست است. در اینجا مثالهایی که این مفهوم را تشریح می کنند آورده شده است.

```
if (myChoice == 'A' and myAge < 25)
```

بصورت زیر ارزیابی می شود:

```
if ((myChoice == 'A') and (myAge < 25))
```

فرض کنید x=8 و y=49 و z=1 باشد.

```
if (x < 7 && y > 50 || z < 2)
```

اگر بصورت زیر ارزیابی شود درست (true) است:

```
if (((x < 7) && (y > 50))   ||   (z < 2))
```

و اگر بصورت زیر ارزیابی شود نادرست (false) است:

```
if ((x < 7)   &&   ((y > 50) || (z < 2)) which is FALSE.
```

در اینجا نکات پایانی برای نتیجه گرفتن از این درس گفته می شود. اول، با وجو د اینکه شما در مورد ترتیب تقدم یک عملگر مطمئن هستید، از پرانتزها بصورت صریح استفاده کنید. این به افزایش قابلیت خوانایی برنامه و اجتناب از خطاها کمک می کند. دوم، در اینجا چیزی مانند بستنی چای سبز است و من به شما توصیه می کنم که آن را نخرید!

پیاده سازی بدون عملگرهای منطقی

```
if (myLetter == 'A') {
    vowelCnt++;
}
else if (myLetter == 'E') {
    vowelCnt++;
}
else if (myLetter == 'I') {
    vowelCnt++;
}
else if (myLetter == 'O') {
    vowelCnt++;
}
else if (myLetter == 'U') {
    vowelCnt++;
}
else {
    constanantCnt++;
}
```

پیاده سازی با عملگرهای منطقی

```
if ((myLetter == 'A') ||
    (myLetter == 'E') ||
    (myLetter == 'I') ||
    (myLetter == 'O') ||
    (myLetter == 'U')) {
    vowelCnt++;
```

```
}
else {
    constantantCnt++;
}
```

ترجمه: محمد صافدل
منبع: http://cplus.about.com