

آموزش C++ - درس ۵- پردازش شرطی: قسمت اول

دستورات If/Else و عملگرهای مربوطه

به سری آموزشهای C++ خوش آمدید. این درس دستورات شرطی را معرفی می کند. در درسهای قبل، همه کد موجود در مثال اجرا می شد، از خط اول برنامه تا آخر، هر دستور به ترتیب محل آن در کد منبع اجرا می شد. این ممکن است در برخی برنامه ها درست باشد اما در بعضی دیگر باید برای انتخاب اینکه کدام دستورها اجرا شوند راهی باشد. پردازش شرطی با استفاده از اجازه به کاربردن منطق ساده یا بررسی تصمیم گیری برای اینکه کدام بلاک از کد اجرا شود، سودمندی برنامه ها را توسعه می دهد. در این درس برای تشریح نحوه استفاده از اجرای شرطی مثال ساده ای (بازی حدس زدن) نوشته خواهد شد.

دستور if برای اجرای شرطی بلاکی از کدها با بررسی درستی شرط استفاده می شود. اگر شرط درست باشد بلاک کد اجرا می شود در غیر این صورت از روی آن می پرد.

تعریف: if یک کلمه رزرو شده در C/C++ است که امکان اجرای شرطی کد را می دهد.
مثال:

```
If (myAnswer=='Y')
{
    Printf("DO This\n");
}
```

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

int main()
{
    int number = 5;
    int guess;

    cout << "I am thinking of a number between 1 and 10" << endl;
    cout << "Enter your guess, please " << endl;
    cin >> guess;
    if (guess == number)
    {
        cout << "Incredible, you are correct" << endl;
    }

    return 0;
}
```

مثال بالا را کمپایل و اجرا کنید. "==" عملگر رابطه ای نامیده می شود. عملگرهای رابطه ای == و != و > و < و <= برای مقایسه دو عملوند بکار می روند. برنامه کار می کند اما آن لزوما پیشرفتهایی لازم دارد. اگر کاربر عدد ۵ را به عنوان انتخابش وارد کند، یک پیغام مناسب دریافت می کند: "Incredible, you are correct". اما اگر کاربر یک انتخاب اشتباه داشته باشد چه اتفاقی می افتد؟ هیچ. نه پیغامی، نه اشاره ای، هیچی. خوشبختانه برای کاربر برنامه ما C++ یک راه حل دارد.

دستور else راهی فراهم می کند تا بلاکی از کد در صورت درست نبودن شرط If اجرا شود.

تعریف: else یک کلمه رزرو شده در C/C++ است که به همراه if امکان اجرای شرطی کد را به ما می دهد.
مثال:

```
if (myAnswer=='Y')
{
    printf("DO This\n");
}
else
{
    printf("DO That\n");
}
```

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

int main()
{
    int number = 5;
    int guess;

    cout << "I am thinking of a number between 1 and 10" << endl;
    cout << "Enter your guess, please ";
    cin >> guess;
    if (guess == number)
    {
        cout << "Incredible, you are correct" << endl;
    }
    else
    {
        cout << "Sorry, try again" << endl;
    }

    return 0;
}
```

این پیشرفت بزرگی است. بدون توجه به درست یا غلط بودن حدس، کاربر پیغامهایی دریافت می کند. اما اجازه دهید کارهایی در برنامه انجام دهیم تا به یک بازی واقعی شبیه تر شود. وقتی که این بازی لذت بخش را به همراه دوستان و خانواده انجام می دهید، موقعی که یک حدس اشتباه زده شود ما چه باید بگوییم؟ بالاتر یا پایینتر. C++ یک ساختار if/else if دارد که می تواند برای چنین موقعیتهایی استفاده شود.

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

int main()
{
    int number = 5;
    int guess;

    cout << "I am thinking of a number between 1 and 10" << endl;
    cout << "Enter your guess, please ";
    cin >> guess;
    if (guess == number)
    {
        cout << "Incredible, you are correct" << endl;
    }
    else if (guess < number)
    {
        cout << "Higher, try again" << endl;
    }
    else // guess must be too high
    {
        cout << "Lower, try again" << endl;
    }
    return 0;
}
```

به این نکته جالب توجه کنید که برخلاف دیگر زبانهای برنامه نویسی، if else لغت کلیدی در C++ نیست. برای مثال، زبان Perl دارای لغت کلیدی elseif است. ساختار if/else if از دستورات if و else ساخته شده است. برای دیدن این موضوع کد بالا را می توان به شکل زیر بازنویسی کرد:

```
if (guess == number)
{
    cout << "Incredible, you are correct" << endl;
}
else
    if (guess < number)
```

```

{
    cout << "Higher, try again" << endl;
}
else // guess must be too high
{
    cout << "Lower, try again" << endl;
}

```

این کد با کد قبلی یکی است. فقط فاصله ها برای تشریح چگونگی دستورات if/else if ایجاد شده، تغییر کرده اند.

تمرین

یک بازی بنویسید که اجازه دهد تا کاربر تاریخ تولد شما را حدس بزند. منطقی را به برنامه تان اضافه کنید تا حدسها را بین ۱ و ۳۱ محدود کند، همانطور که روزهای یک ماه محدود به آن هستند. اگر حدس خارج از این محدوده است پیامهای خطای مناسبی را صادر کنید. اگر حدس اشتباه بود، منطق و پیامهایی را برای راهنمایی کاربر به سمت پاسخ صحیح اضافه کنید مانند "Higher" یا "Lower".

خلاصه ای از عملگرهای رابطه ای

عملگر	توضیح	مثال	ارزش
==	مساوی	5 == 4	FALSE
		5 == 5	TRUE
!=	نامساوی	5 != 4	TRUE
		5 != 5	FALSE
>	بزرگتر	5 > 4	TRUE
		5 > 5	FALSE
>=	بزرگتر یا مساوی	5 >= 4	TRUE
		5 >= 5	TRUE
<	کوچکتر	5 < 4	FALSE
		5 < 5	FALSE
<=	کوچکتر یا مساوی	5 <= 4	FALSE
		5 <= 5	TRUE

حل تمرین:

```

#include <iostream>
using namespace std;

int main()
{
    int myBirthday = 13;
    int guess;

    cout << "Please guess the day of my birth, from 1 to 31" << endl;
    cout << "Enter your guess, please ";
    cin >> guess;
    // Test for out of range guesses first
    if (guess < 0)
    {
        cout << "Months don't have negative days, Einstein" << endl;
    }
    else if (guess > 31)
    {
        cout << "Pretty long month, genius" << endl;
    }
    else if (guess == myBirthday)
    {

```

```

    cout << "Incredible, you are correct" << endl;
}
else if (guess < myBirthday)
{
    cout << "Higher, try again" << endl;
}
else // Guess is in range and not greater than or equal
{
    cout << "Lower, try again" << endl;
}
return 0;
}

```

نکته: استفاده از آکولادها در if

C/C++ دو حالت برای دستورات تکی در if در نظر گرفته است، یکی با آکولادها ({}) و دیگری بدون آن.

```

if (myPocketChange == 1000)
{
    printf("Do'in Fine");
}

```

یا این حالت:

```

if (myPocketChange == 1000)
    printf("Do'in Fine");

```

از کدام باید استفاده کنیم؟ بطور کلی اگرچه حالت دوم نیاز به تایپ کمتری دارد ولی از حالت اول یعنی استفاده از آکولادها استفاده کنید. چرا؟ وقتی از دستورات if تودرتو استفاده می کنید یا از حالت دنداندار استفاده نمی کنید، پیدا کردن غلطهایی که ممکن است رخ دهند بسیار مشکل است.

```

myEggs = 10;
if (myEggs == 12)
    printf("You Have a Dozen");
    printf("Prepare the Omelet!!");
strcpy(MyCheese,"CHEDDAR");

```

خروجی این برنامه به شکل زیر است:

```
Prepare the Omelet!!
```

تعجب کردید؟ قبل از اینکه به دنبال پیدا کردن تخم مرغ بروید! به این نکته توجه کنید که کمپایلر به دنداندارها توجه نمی کند. فقط اولین دستور print در قسمت شرطی قرار دارد. دومین دستور print قسمتی از دستور if به حساب نمی آید. با دنداندار گذاری صحیح همان کد بصورت زیر نوشته می شود:

```

myEggs = 10;
if (myEggs == 12)
    printf("You Have a Dozen");
printf("Prepare the Omelet!!");
strcpy(MyCheese,"CHEDDAR");

```

با استفاده از آکولادها باز هم کد خواناتر می شود:

```

myEggs = 10;
if (myEggs == 12)
{
    printf("You Have a Dozen");
}
printf("Prepare the Omelet!!");
Strcpy(MyCheese,"CHEDDAR");

```