

Introduction to C Programming

Module - 4

Conditional Statement

সিদ্ধান্ত গ্রহণের জন্য প্রোগ্রামিং এ সচরাচর কন্ডিশনাল(conditional) স্টেটমেন্ট ব্যবহৃত হয়।

সি প্রোগ্রামিং এ তিন ধরনের if কন্ডিশনাল(conditional) স্টেটমেন্ট রয়েছে:

1. ইফ-if
2. ইফ...এলস- if...else
3. ইফ...এলসইফ...এলস - if...elseif...else

সি **if** স্টেটমেন্ট সিনট্যাক্স

if (testExpression)

{

// True হলে এই কোড এক্সিকিউট হবে।

}

এখানে if প্রথমে testExpression-কে মূল্যায়ন করে।

testExpression যদি True(শূন্য না) হয় তাহলে if ব্লকের মধ্যে অবস্থিত স্টেটমেন্ট/স্টেটমেন্টসমূহ সম্পাদিত হবে।

testExpression যদি False(শূন্য) হয় তাহলে if ব্লকের মধ্যে অবস্থিত স্টেটমেন্ট/স্টেটমেন্টসমূহ এড়িয়ে যাবে।

সি **if...else** স্টেটমেন্ট

কন্ডিশন True হলে কিছু কোড এবং False হলে অন্য কিছু কোড সম্পাদন(execution) এর জন্য সি if...else স্টেটমেন্ট ব্যবহৃত হয়।

if...else সিনট্যাক্স

if (testExpression) {

// True হলে এই কোড এক্সিকিউট হবে।

}

else {

// False হলে এই কোড এক্সিকিউট হবে।

}

টেস্ট এক্সপ্রেশন True হলে if স্টেটমেন্টের কোড ব্লক এক্সিকিউট হবে এবং else স্টেটমেন্টের কোড ব্লকে এড়িয়ে যাবে।

টেস্ট এক্সপ্রেশন false হলে else স্টেটমেন্টের কোড ব্লক সম্পাদিত হবে এবং if স্টেটমেন্টের কোড ব্লকে এড়িয়ে যাবে।

নেস্টেড **if...else** স্টেটমেন্ট

কন্ডিশন True/False যাই হোক না কেন আমরা কোড এক্সিকিউট করতে চাই অর্থাৎ সিদ্ধান্ত গ্রহণ করতে চাই সেক্ষেত্রে if...else স্টেটমেন্ট ব্যবহৃত হয়। কিন্তু মাঝে মাঝে সিদ্ধান্ত গ্রহণ দুই এর অধিক কন্ডিশনের উপর নির্ভর করতে পারে। দুই এর অধিক এক্সপ্রেশনকে টেস্ট করে কোড এক্সিকিউট করার জন্য নেস্টেড if...else স্টেটমেন্ট ব্যবহৃত হয়।

নেস্টেট **if...else** সিনট্যাক্স

```
if (testExpression1)
{
    //testExpression1 True হলে এই কোড এক্সিকিউট হবে।
}
else if(testExpression2)
{
    // testExpression1 false এবং testExpression2 True হলে এই কোড ব্লক এক্সিকিউট হবে।
}
else if (testExpression3)
{
    //testExpression1 ও testExpression2 false এবং testExpression3 True হলে এই কোড ব্লক এক্সিকিউট হবে।
}

else
{
    // সকল testExpression false হলে এই কোড ব্লক এক্সিকিউট হবে।
}
```

সি রিলেশনাল অপারেটর

সি প্রোগ্রামিং এ আমরা প্রায়ই রিলেশনাল বা সম্পর্কযুক্ত কাজ করে থাকি। এই সমস্ত কাজ সম্পাদন করার জন্য আমরা রিলেশনাল অপারেটর ব্যবহার করি। অর্থাৎ এই সকল অপারেটর দিয়ে দুটি অপারেন্ডের মধ্যে সম্পর্ক যাচাই করা হয়। আরো সহজভাবে বললে একটি সংখ্যা আরেকটি সংখ্যা থেকে ছোট না বড়, সমান না অসমান, জোড় না বেজোড়, সত্য না মিথ্যা এই সমস্ত সূক্ষ্ম বিষয় যাচাই করার জন্য এই সকল অপারেটর ব্যবহৃত হয়। রিলেশন সত্যি হলে সে 1 রিটার্ন করে, রিলেশন মিথ্যা হলে সে 0 রিটার্ন করে।

রিলেশনাল অপারেটর কি কি এবং তাদের কাজের ধরণ সি প্রোগ্রামে কেমন হয় তা জানতে নিচের টেবিলটি লক্ষ্য করুন-

Relational Operators

| Operators | Meaning | Example | Result |
|-----------|----------------------------------|------------|--------|
| < | Less than | 5<2 | False |
| > | Greater than | 5>2 | True |
| <= | Less than or equal to | 5<=2 | False |
| >= | Greater than or equal to | 5>=2 | True |
| == | Equal to | 5==2 | False |
| != | Not equal to | 5!=2 | True |
| === | Equal value and same type | 5 === 5 | True |
| | | 5 === "5" | False |
| ! == | Not Equal value or Not same type | 5 ! == 5 | False |
| | | 5 ! == "5" | True |

Evaluate: মূল্যায়ন, সংখ্যা নির্ণয় করা।

Boolean

লজিক্যাল চলক এবং লজিক্যাল অপারেশনসমূহের মাধ্যমে গঠিত গণিতকে বুলিয়ান বীজগণিত বা বুলিয়ান অ্যালজেব্রা বলে। এটি যৌক্তিক/লজিক্যাল সত্য এবং মিথ্যার উপর ভিত্তি করে তৈরি। সত্য এবং মিথ্যাকে যথাক্রমে 0 এবং 1 দ্বারা প্রকাশ করা হয়

Logical Operator

C প্রোগ্রামিং এ দুটি Logical Operator রয়েছে। তাদের নিচে দেওয়া হল:

| Operator | Meaning |
|----------|---------|
| && | And |
| | Or |

&& কে বলা হয় Logical and এবং || কে বলা হয় Logical Or.

&& (পড়া হয় And operator)

মনে করি x,y,z তিনটি চলক। এখন (x<y)&& (y<z) হচ্ছে একটি expression. এখন এর মান সত্য হবে যদি (x<y) এবং (y<z) সত্য হয়। (x<y) এবং (y<z) এর যে কোন একটি মিথ্যা হলে (x<y)&& (y<z) এর মান মিথ্যে হবে।

||(পড়া হয় Or Operator)

মনে করি x,y,z তিনটি চলক। এখন (x<y)|| (y<z) হচ্ছে একটি expression। এখন এর মান সত্য হবে যদি (x<y) এবং (y<z) সত্য হয়। অথবা (x<y) এবং (y<z) এর যে কোন একটি সত্য হয়। (x<y) এবং (y<z) দুটি একসাথে মিথ্যা হলে (x<y)|| (y<z) এর মান মিথ্যে হবে।

Ternary Operator

টারনারী অপারেটরকে কন্ডিশনাল লজিক (if else) এর ভাই বলা চলে। কারণ, আমরা কন্ডিশনাল লজিক (if else) ব্যবহার করার পরিবর্তে টারনারী অপারেটর ব্যবহার করে অনেক সহজে ও অনেক কম কোড লিখে একই কাজ করতে পারি।

expression-1 ? expression-2 : expression-3

উপরের expression-1 এর জায়গায় যে কন্ডিশন বা শর্ত দিবেন তা লিখতে হবে তারপর expression-2 ও expression-3 এর জায়গায় যেকোনো value বা variable বা statement বা কোন mathematical expression দিতে পারেন। এখন যদি expression-1 এর জায়গায় যে কন্ডিশন বা শর্ত দিবেন সেই কন্ডিশন বা শর্ত যদি সত্যি হয় তাহলে ? চিহ্ন এর পরের অংশ বা expression-2 execute হবে আর যদি কন্ডিশন বা শর্ত সত্যি না হয় তাহলে : চিহ্ন এর পরের অংশ বা expression-3 execute হবে।