

সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

একটি প্রোগ্রামিং ল্যাবুরেজের উদ্দেশ্য শুধু গাণিতিক হিসাব-নিকাশের মধ্যেই সীমাবদ্ধ নয়, বরং তা তৈরি করা হয় মানুষের কাজকে সহজ করার জন্য। তাই একটি প্রোগ্রামিং ল্যাবুরেজের মধ্যে দ্রুত করা হয় বিভিন্ন সিদ্ধান্তমূলক কাজ করার কলাকৌশল। বিভিন্ন শর্ত তথা কন্ডিশনাল অপারেশন একটি প্রোগ্রামিং ল্যাবুরেজের অন্যতম উদ্দেশ্য। আসলে একটি ল্যাবুরেজের জন্য ম্যাথমেটিক্যাল অপারেশন এবং কন্ডিশনাল অপারেশন সমান ভূমিকা পালন করে।

সি দিয়ে বিভিন্ন সিদ্ধান্তমূলক কাজ এবং বিভিন্ন শর্তের ওপর নির্ভর করে কোনো কাজের পুনরাবৃত্তি করা যায়। এই দুই ধরনের কাজকে একত্রে বলা হয় স্টেটমেন্ট নিয়ন্ত্রণ। স্টেটমেন্ট নিয়ন্ত্রণের জন্য সিন্টাক্সের সুবিধা আছে, যা যেসব পদ্ধতি অবলম্বন করে কাজটি করা হয় তাকে কন্ট্রোল স্টেটমেন্ট বলে। এ লেখায় কন্ট্রোল স্টেটমেন্ট সম্পর্কে বিস্তারিত আলোচনা করা হয়েছে। তাই প্রথমে কিছু মৌলিক বিষয় সম্পর্কে জেনে নিই।

স্টেটমেন্ট

আপে কিছু নির্দিষ্ট ধরনের এক্সপ্রেশন দিয়ে আলোচনা করা হয়েছে। তা ছাড়া বিভিন্ন লাইব্রেরি ফাংশন নিয়েও (যেমন : printf(), scanf() ইত্যাদি) সাধারণ দেয়া হয়েছে। সি-তে কোনো কোডের শেষ বোঝাতে সেন্সিটিভিটি (:) ব্যবহার করা হয়। একটি কম্পিট এক্সপ্রেশন যখন সেন্সিটিভিটি দিয়ে শেষ করা হয় তখন সিন্টাক্স তাকে স্টেটমেন্ট বলে। স্টেটমেন্ট সাধারণত দুই রকমের হয়ে থাকে। যেমন :

সিম্পল স্টেটমেন্ট : একটিমাত্র এক্সপ্রেশন বা ফাংশন দিয়ে যে স্টেটমেন্ট গঠিত হবে, তা হলো সিম্পল স্টেটমেন্ট।

কম্পাউন্ড স্টেটমেন্ট/কোড ব্লক : একাধিক স্টেটমেন্টকে যখন দ্বিতীয় বন্ধনী '}' মধ্যে লেখা হবে, তখন তাকে কম্পাউন্ড স্টেটমেন্ট বা কোড ব্লক বলে। সিম্পল এবং কম্পাউন্ড স্টেটমেন্টের মধ্যে বিশেষ কোনো পার্থক্য নেই, শুধু কোড বা কার্যক্রমের ভিন্নতা দেখা যায়।

```
printf("hello!"); //সিম্পল স্টেটমেন্ট
{ //কম্পাউন্ড স্টেটমেন্ট
    int x=1;
    printf("%d",&x);
    x++;
}
```

কম্পাউন্ড স্টেটমেন্ট সম্পর্কে বিশেষ কিছু কথা : একটি সিম্পল স্টেটমেন্টকে যদি দ্বিতীয় বন্ধনী দিয়ে আবদ্ধ করে কোনো কোড ব্লক তৈরি করা হয়, তাহলেও সেটি কম্পাউন্ড স্টেটমেন্ট হিসেবে বিবেচিত হবে। প্রোগ্রামে কোড ব্লকের

ভেতরে-বাইরে সব জায়গায় যেকোনো ভেরিয়েবল ডিক্লার করা সম্ভব। কিন্তু সে থেকে ভেরিয়েবলের স্কোপ ভিন্ন হবে। কোনো কোড ব্লকের ভেতরে কোনো ভেরিয়েবল ডিক্লার করলে তার স্কোপ ওই কোড ব্লকের মধ্যেই সীমাবদ্ধ থাকবে।

```
int x;
x=42;
{
    int x=53;
    printf("%d",&x);
}
printf("%d",&x);
```

এখানে একই ভেরিয়েবল x-কে দুইবার ডিক্লার করা হয়েছে। কিন্তু ভেরিয়েবল দুটির স্কোপ ভিন্ন, তাই প্রোগ্রামে এখানে কোনো এরর দেখাবে না। প্রথম x-এর স্কোপ কোড ব্লকের বাইরে, তাই ব্লকের বাইরে যেকোনো জায়গায় x-এর মান প্রিন্ট করতে চাইলে ৪২ আউটপুট আসবে, কিন্তু কোড ব্লকের ভেতরে x-কে প্রিন্ট করতে চাইলে ৫৩ প্রিন্ট হবে, কারণ দ্বিতীয় x-এর স্কোপ কোড ব্লকের ভেতরে। কিন্তু কোড ব্লকের বাইরে দ্বিতীয়বার x-কে ডিক্লার করলে চাইলে এরর দেখাবে।

এখন এসব স্টেটমেন্টকে কিভাবে আমাদের ইচ্ছানুযায়ী ব্যবহার করা যায় সে বিষয়ে অর্থাৎ কন্ডিশনাল স্টেটমেন্টের কাজ সম্পর্কে আলোচনা করা হয়েছে। একটি কথা বলে রাখা ভালো, কম্পিউটার শুধু দ্রুত হিসাব-নিকাশ করতে পারে বলেই যে এত বেশি ব্যবহার হয় তা নয়, বরং আমাদের ইচ্ছানুযায়ী সিদ্ধান্ত নিতে পারে বলেই এর প্রয়োজনীয়তা এত বেশি। এর মান হলো প্রোগ্রামিংয়ে কন্ডিশনাল স্টেটমেন্টের গুরুত্ব বেশি। যেমন : বিভিন্ন সফটওয়্যারে অনেক সময়ই কিছু কমান্ড দেখা যায়, যেমন : Do you want to quit? (y/n) যেখানে y চাপলে প্রোগ্রাম থেকে বের হয়ে যাবে আর n চাপলে আবার প্রোগ্রামে ফিরে যাবে। এ ধরনের কাজই হলো কন্ডিশনাল অপারেশন।

সি-তে কন্ডিশনাল অপারেশনের জন্য যেসব কিওয়ার্ড ব্যবহার করা হয় তাদেরকে কাজের সুবিধার্থে দুই ভাগে ভাগ করা যেতে পারে। যেমন : কন্ডিশনাল কিওয়ার্ড, যাদের কাজ হলো কখন কোন ধরনের কন্ডিশন ট্রিক করতে হবে তা নির্ধারণ করে দেয়া, আরেকটি হলো লুপিং কিওয়ার্ড যাদের কাজ হলো কোন কোড ব্লককে বিভিন্ন নিয়মানুসারে পুনরাবৃত্তি করা।

- কন্ডিশনাল কিওয়ার্ড : if, else, switch, case, break, default
- লুপিং কিওয়ার্ড : while, do, for, goto

কন্ডিশনাল অপারেশনের কর্মপদ্ধতি

একটি কন্ট্রোল স্টেটমেন্ট বা কোনো কন্ডিশনাল অপারেশনের সাধারণত দুটি অংশ

থাকে। একটি হলো কন্ডিশন অংশ এবং অন্যটি হলো স্টেটমেন্ট অংশ। কন্ডিশন অংশে একটি এক্সপ্রেশন ব্যবহার করা হয়, যার মান থেকে প্রোগ্রাম বুঝতে পারে যে কন্ডিশনটি সত্য না মিথ্যা। শূন্য ছাড়া অন্য যেকোনো মানকে প্রোগ্রাম সত্য বলে ধরে নেয়। যেমন : a=2; b=3; হলে যদি কন্ডিশনের এক্সপ্রেশন হিসেবে a<b এবং a+b ব্যবহার করা হয়, তবে কন্ডিশনের মান হবে 1 তথা সত্য। আর যদি a>b এবং (a-b+1) ব্যবহার করা হয় তাহলে কন্ডিশনের মান হবে 0 তথা মিথ্যা। এভাবে সরাসরি রিলেশনাল অপারেটর দিয়ে অথবা সাধারণ কোড কন্ডিশনাল এক্সপ্রেশন দিয়ে কোনো কন্ডিশনের মান নির্ধারণ করা যায়। আবার কন্ডিশনের মান হিসেবে যদি সরাসরি কোনো সংখ্যা বা মান বা ভেরিয়েবল দেয়া হয় তাহলে সরাসরি সেই মানের ওপর ভিত্তি করে কন্ডিশনের মান ট্রিক করা হয়। যেমন : কোনো কন্ডিশনের এক্সপ্রেশন হিসেবে যদি শুধু 0 ব্যবহার করা হয় তাহলে সেটি সত্য হবে, কারণ 0 হলো একটি অশূন্য (নি-নিয়ম) সংখ্যা।

এখন বিভিন্ন কন্ডিশনাল/লুপিং কিওয়ার্ড ব্যবহার করে কিভাবে কন্ডিশনাল স্টেটমেন্ট তৈরি করতে হয় সেটি আলোচনা করা হবে। তবে তার আগে বলে রাখা ভালো কোনো কিওয়ার্ডের সাথে ব্যবহার হওয়া স্টেটমেন্ট বা কোড ব্লককে ওই কিওয়ার্ড সফটওয়্যারে স্টেটমেন্ট বলে। যেমন : if statement, while statement ইত্যাদি।

if স্টেটমেন্ট : কন্ডিশন দিয়ে কাজ করার জন্য যাপকভাবে ব্যবহার হওয়া পর্যন্ত হলো if স্টেটমেন্ট। প্রথমে একটি বাক্য লক্ষ করা যাক : যদি বৃষ্টি হয়, তাহলে আমরা যাব না। এখানে একটি কাজ হলে আরেকটি কাজ করা হবে। অর্থাৎ প্রথম কাজের ওপর দ্বিতীয় কাজ হবে কি হবে না তা নির্ভর করবে। সি ল্যাবুরেজের এ ধরনের কন্ডিশন দিয়ে কাজ করার জন্য সাধারণত if স্টেটমেন্ট ব্যবহার হয়। আর if স্টেটমেন্টের সাথে কন্ডিশন হিসেবে একটি এক্সপ্রেশন থাকে, যা নির্ধারণ করে যে কোন কাজটি সম্পন্ন করতে হবে। প্রোগ্রামে if-কে তিনভাবে ব্যবহার করা যায়। যেমন : সাধারণ if হিসেবে, if-else স্টেটমেন্ট হিসেবে অথবা else if চেইন হিসেবে।

সাধারণ if-এর কাজ একদমই সহজ। একটিমাত্র কন্ডিশন দেয়া থাকে এবং সেটি সত্য হলে কোনো কাজ সম্পন্ন হবে, না হলে সম্পন্ন হবে না। যেমন : int age; scanf("%d",&age); if(age==18) printf("You are mature."); if(age<=18)

(যদি কমে ১৮ পূর্ণ হবে)



প্রোগ্রামিং সি/সি++

(৯৩ পৃষ্ঠার পর)

```
printf("You are immature.");
```

এখানে প্রতিটা if-এর নিচে যে প্রিন্ট করার কমান্ড দেয়া হয়েছে তা হলো কাজ, আর if-এর ডান পাশে প্রথম বন্ধনীর মাঝে যা আছে তা হলো কন্ডিশন। এই কন্ডিশনটিই নির্ধারণ করে যে সফটওয়্যার if স্টেটমেন্টটি কার্যকর হবে কি না।

if—else স্টেটমেন্টটি একটু জটিল। আগের if-এ কন্ডিশনটি সত্য হলে কোনো কাজ হবে, না হলে কিছুই হবে না। আর এবার কন্ডিশন সত্য হলে কোনো কাজ হবে আর মিথ্যা হলে অন্য কোনো কাজ হবে। যেমন :

```
If(age>=18)
printf("You are mature.");
else
printf("You are immature.")
```

এখানে age-এর মান যদি 1৮ বা তার বেশি হয় তাহলে প্রথম লাইনটা প্রিন্ট করবে। আর যদি মান 1৮-এর কম হয় তাহলে প্রোগ্রাম নিজে থেকেই দ্বিতীয় লাইনটা প্রিন্ট করবে।

else if চেইনের কাজ তুলনামূলক জটিল। যদি অনেকগুলো কন্ডিশনের সাথে অনেকগুলো কাজ করার প্রয়োজন হয় তাহলে if else চেইন ব্যবহারের প্রয়োজন পড়ে। যেমন :

```
if(age==50)
```

```
printf("You are old.");
else if(age>=25)&&(age<50)
printf("You are young");
else if(age>=18)&&(age<25)
printf("You are mature");
else if(age>=10)&&(age<18)
printf("You are a boy");
else if(age<10)&&(age>=0)
printf("You are a child");
else
printf("You are not born!!");
```

ওপরের else if চেইনে কয়েকটি বিষয় লক্ষণীয়। if-এর সাথে যেমন কোনো কন্ডিশন ব্যবহার হয়, তেমনি else if-এর সাথেও কন্ডিশন থাকে। তবে শুধু else-এর সাথে কোনো কন্ডিশন থাকে না। কারণ এটি হলো ডিফল্ট অপারেশন। অর্থাৎ যখন কোনো কন্ডিশনই সত্য হবে না, তখন এই ডিফল্ট অপারেশনটি কার্যকর হবে। if বা else বা else if কিওয়ার্ডের পর কোনো কন্ডিশন থাকুক অথবা না থাকুক কোনো সেমিকোলন দেয়া যাবে না। তবে তাদের অধীনে যেসব স্টেটমেন্ট থাকে তাদের শেষে সাধারণ নিয়মানুযায়ী সেমিকোলন ব্যবহার করতে হবে। ওপরের প্রতিটি else if-এর কন্ডিশনগুলো একটু জটিল মনে হতে পারে। কিন্তু তা মোটেও জটিল নয় বরং এখানে দুটি কন্ডিশন একত্রে ব্যবহার করা হয়েছে। পরে দুটি কন্ডিশনকে && (আন্ড) দিয়ে যুক্ত করা হয়েছে। এখানে & ব্যবহার না করে && ব্যবহার করা হয়েছে, কারণ & হলো বিটওয়াইজ অপারেটর এবং && হলো

বাইনারি অপারেটর। কেউ যদি লো লেভেল প্রোগ্রামিং করতে চান, তাহলে সে ক্ষেত্রে বিটওয়াইজ অপারেটর ব্যবহার করার দরকার পড়ে, কিন্তু সাধারণ প্রোগ্রামিংয়ে সবসময় বাইনারি অপারেটর ব্যবহার করতে হয়। কন্ডিশন && দিয়ে যুক্ত করার মানে হলো দুটো কন্ডিশনই যদি সত্য হয় তাহলে পুরো কন্ডিশনটি সত্য হবে। যেমন : ওপরের উদাহরণটিতে তৃতীয় কন্ডিশনের নিকে বেয়াল করলে বোকা যাবে, ওই কন্ডিশনটি তখনই সত্য হবে যখন age-এর মান 1৮-এর সমান অথবা বড় হবে এবং age-এর মান ২৫-এর কম হবে, অর্থাৎ দুটো কন্ডিশনই সত্য হবে। কিন্তু এমন যদি হতো যে (age>=18) এই কন্ডিশনটি শুধু সত্য হতো কিন্তু (age<25) এই কন্ডিশনটি মিথ্যা হতো তখন ((age>=18)&&(age<25)) এই পুরো কন্ডিশনটি মিথ্যা হবে। যদি যেকোনো একটি কন্ডিশনের সত্য হওয়ার জন্য পুরো কন্ডিশনটি সত্য করতে হয় তাহলে :: (অথবা) এই অপারেটরটি ব্যবহার করতে হবে। সুতরাং তখন পুরো কন্ডিশনটি হবে ((age>=18)||(age<25))

সি ল্যাঙ্গুয়েজে কন্ডিশনাল/লুপিংয়ের জন্য আরও অনেক ধরনের উপায় আছে। সাধারণত if ব্যবহার করা হয়, তবে মাঝেমাঝে বিভিন্ন বড় কাজের জন্য অন্য পদ্ধতিগুলোও ব্যবহার করা হয়। পরে বাকি পদ্ধতিগুলো এবং কিওয়ার্ডগুলোর কাজ সম্পর্কে আলোচনা করা হবে। ■

কিতাবখানা : wahid_cseanst@yahoo.com