

# সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

যেকোনো প্রোগ্রামিং ল্যাঙ্গুয়েজের কাজ করার বেসিক হলো বিভিন্ন হিসাব-নিকাশের মাধ্যমে বিভিন্ন কাজ সম্পাদন করা। আর একসব হিসাব-নিকাশের জন্য দরকার কিছু অপারেটর। সি একটি প্রোগ্রামিং ল্যাঙ্গুয়েজ যা দিয়ে সাধারণ গণিতের বাইরেও অনেক রকম হিসাব-নিকাশ করা সম্ভব। আজকে প্রোগ্রামিং ল্যাঙ্গুয়েজ সি-তে কী ধরনের অপারেটর রয়েছে এবং তা নিয়ে কী ধরনের কাজ করা সম্ভব তা তুলে ধরা হয়েছে।

অপারেটর বলতে বিশেষ ধরনের কিছু ক্যারেকটার বোঝানো হয়, যা দিয়ে কোনো সম্পর্ক স্থাপন করা যায়। ক্যালকুলেটরে যেমন-যোগ, বিয়োগ, গুণ, ভাগ ইত্যাদি করার জন্য বিশেষ বিশেষ অপারেটর আছে তেমনি সি ল্যাঙ্গুয়েজও বিভিন্ন হিসাব-নিকাশের জন্য অনেক রকমের অপারেটর আছে।

সি-তে শুধু যে সাধারণ হিসাব-নিকাশ করা সম্ভব তা নয়, বরং বিভিন্ন লজিক্যাল কাজও করা সম্ভব। লজিক্যাল কাজ বলতে বিভিন্ন ধরনের শর্ত নিয়ে কাজ করা বোঝায়। যেমন 'আজকে বৃষ্টি হচ্ছে রহিম বসায় থাকবে' এই বাক্যটিতে একটি শর্ত কাজ করছে। সি-তে এ ধরনের বিভিন্ন শর্ত নিয়ে তথ্য লজিক্যাল অপারেশন করা সম্ভব। কানের ভিত্তিতে সি-এর সব অপারেটরকে মোট ৬ ভাগে ভাগ করা যায়। যেমন :

আরিথমেটিক অপারেটর : +, -, \*(multiplication), /(division), %(modulus)

রিলেশনাল অপারেটর : >, <, >=, <=,

!(not equal), ==(equal)

লজিক্যাল অপারেটর :

!(not), &&(and), ||(or)

বিটওয়াইজ অপারেটর : ~(complement), &(and), |(or), ^(XOR), <<(left shift), >>(right shift)

অ্যাসাইনমেন্ট অপারেটর : +=

(assignment), ++(increment), --

(decrement), +=(add and assign), -=(subtract and assign), \*=

(multiply and assign), /=(divide and assign), %=

(mod and assign), |=

(bitwise OR and assign), &=

(bitwise AND and assign), ^=

(bitwise XOR and assign), <<<

(LEFT SHIFT and assign), >>>

(RIGHT SHIFT and assign)

সিলেকশন অপারেটর : ?:(if-then-else)

এতগুলো অপারেটর দেখে স্বাভাবিকভাবেই ভয় লাগতে পারে, কিন্তু মজার ব্যাপার হলো

সাধারণ সি প্রোগ্রামিংয়ে সিলেকশন অপারেটর এবং বিটওয়াইজ অপারেটর (সব বিটওয়াইজ অপারেটর যেমন অ্যাসাইনমেন্ট অপারেটরের মাঝেও কিছু বিটওয়াইজ অপারেটর আছে) ব্যবহারের প্রয়োজন পড়ে না, কারণ এটি দিয়ে সরাসরি বিট নিয়ন্ত্রণ করা হয় যা লো লেভেল প্রোগ্রামিংয়ের বৈশিষ্ট্য। তবে অসেকি উই বা খড় সফটওয়্যার বাসাতে এই অপারেটরগুলোর প্রয়োজন পড়তে পারে।

অপারেটর যেসব ভাটা নিয়ে কাজ করে তাদের অপরোধ বলে। যেমন- প্রোগ্রামে যদি  $a+b$  লেখা হয় তাহলে '+' হলো অপারেটর আর  $a$  এবং  $b$  হলো অপারেন্ড। অপারেটরের ওপর ভিত্তি করে অপারেটরকে তিন ভাগে ভাগ করা যায়।

ইউনারি অপারেটর : যেসব অপারেটর শুধু একটি ভাটা বা ভেরিয়েবল নিয়ে কাজ করে তাদের ইউনারি অপারেটর বলে। যেমন : প্রোগ্রামে যদি কোনো ভেরিয়েবল মান এভাবে নির্ধারণ করা হয়  $a=-(3)$ , তাহলে এখানে '-' একটি ইউনারি অপারেটর, কারণ এটি শুধু ৩-কে নিয়ে কাজ করেছে।

বাইনারি অপারেটর : যেসব অপারেটর দুটি ভাটা নিয়ে কাজ করে তাদের বাইনারি অপারেটর বলে। যেমন :  $a=2-5$  লেখা হলে এখানে আবার '-' বাইনারি অপারেটর। কারণ তা ২ এবং ৫ দুজনকে নিয়েই কাজ করেছে।

টারনারি অপারেটর : সি-তে একটি মাত্র অপারেটর আছে যা কি না একই সাথে তিনটি ভাটা নিয়ে কাজ করে। সিলেকশন অপারেটরকেই টারনারি অপারেটর বলে। অপারেটরটি হলো  $?:$  এবং এটি if-else স্টেটমেন্টের সর্বাঙ্গিক রূপ হিসেবে কাজ করে। if-else হলো সর্ব নিয়ন্ত্রণ করার স্টেটমেন্ট যা নিয়ে পরে আলোচনা করা হবে। এই অপারেটরের সিন্টাক্স হলো : (condition)? true;false। এখানে কন্ডিশন তথা শর্তের অংশটুকু সত্য হলে true অংশ কাজ করবে, অন্যথায় false অংশ কাজ করবে। যেমন :  $a=(3>5)?3:5$  এখানে বোঝান হয়েছে যে (3>5) এই শর্তটি যদি সত্য হয় তাহলে  $a$  ভেরিয়েবলের মান হিসেবে ৩ নির্ধারণ হবে, অন্যথায় ৫ নির্ধারণ হবে।

এতকণ দেখানো হলো সি-তে কত ধরনের এবং কি কি অপারেটর আছে। অপারেটরগুলোর কাজ নিয়ে বিস্তারিত আলোচনা করার আগে একটি গুরুত্বপূর্ণ বিষয় নিয়ে আলোচনা করা দরকার।

অপারেটর হিসিডেলপ এবং অ্যাসোসিয়েটিভিটি : প্রোগ্রামে যে সবসময় সাধারণ এক্সপ্রেশন থাকবে তা নয়, জটিল এক্সপ্রেশনও থাকতে পারে। যেমন :  $x=5/4-6*2+81*6*9*(2*-2)/(42+53)$  এটি একটি

জটিল এক্সপ্রেশন। এখানে কোন অপারেটরের আগে কে কাজ করবে সে বিষয়টিকে বলা হয় অপারেটর হিসিডেলপ, আর একই ধরনের অপারেটরগুলো এক্সপ্রেশনের ভদ্র দিক থেকে কাজ করবে না বাম দিক থেকে কাজ করবে সে বিষয়টিকে বলা হয় অপারেটর অ্যাসোসিয়েটিভিটি। যেমন : আমরা জানি কোনো গাণিতিক এক্সপ্রেশনে যোগ/বিয়োগের থেকে গুণ/ভাগের কাজ আগে হয়, তার মানে গুণ/ভাগের হিসিডেলপ যোগ/বিয়োগের থেকে আগে। অ্যাসোসিয়েটিভিটির উদাহরণ প্রোগ্রাম করার সময় বোঝা যাবে। নিচে অপারেটরগুলো নিয়ে বিস্তারিত আলোচনা করা হয়েছে।

আরিথমেটিক অপারেটর : আরিথমেটিক অপারেটরগুলো একদমই সাধারণ। ক্যালকুলেটরের অপারেটরগুলোই আরিথমেটিক অপারেটর। এখানে +, - এই দুটি অপারেটর ইউনারি এবং বাইনারি দু'ভাবেই ব্যবহার করা সম্ভব। আর \*, /, % এই অপারেটর তিনটি বাইনারি অপারেটর হিসেবে ব্যবহার করা হয়। এখানে % নিয়ে ভাগশেষ বোঝান হয়। যেমন :  $a=18*6, b=3*62, c=42*53$ । এখানে  $a=0, b=18, c=42$  হবে। কেননা 18 কে 6 দিয়ে ভাগ করলে ভাগশেষ হয় 0, 3 কে 2 দিয়ে ভাগ করলে ভাগশেষ হয় 1, 42 কে 53 দিয়ে ভাগ করলে ভাগশেষ হবে 42, কারণ এখানে ভাগফলে কোনো পূর্ণ সংখ্যা নেই।

রিলেশনাল অপারেটর : রিলেশনাল অপারেটরগুলো আমাদের কাছে পরিচিত। >, <, >=, <= (greater than or equal), <= (less than or equal) এই অপারেটরগুলো সুবার কাচেই পরিচিত। কিন্তু এখানে নতুন দুটি অপারেটর আছে !=(not equal), ==(equal)। এই অপারেটরগুলো সাধারণ গণিতে লেখা যায় না। যদি প্রোগ্রামে এমন একটি এক্সপ্রেশন থাকে  $a=b$  তাহলে তা বোঝাবে 'a যদি b-এর সমান না হয়'। আর যদি ==(equal) থাকে যেমন  $a=b$  এই এক্সপ্রেশনের মানে হলো 'যদি a এবং b সমান হয়'। এখানে খেয়াল রাখতে হবে '=' এবং '==' এই দুটি অপারেটর কিছ্র আলাদা কাজ করে। যদি  $a=3$  লেখা হয় তাহলে  $a$  ভেরিয়েবলের মান হিসেবে ৩ নির্ধারণ করবে। আর যদি  $a==3$  লেখা হয় তাহলে  $a$  ভেরিয়েবলের মান ৩ কি না তা চেক করবে। তাই '=' কে অ্যাসাইন অপারেটর এবং '==' কে রিলেশনাল অপারেটর বলা হয়।

এখানে আরেকটি কথা বলে রাখা ভালো, সি-তে লজিক্যাল এক্সপ্রেশনও একটি মান রিটার্ন করে। কোনো রিলেশনাল এক্সপ্রেশন যদি সত্য হয়, তাহলে তা 1 রিটার্ন করবে। আর যদি মিথ্যা

হয় তাহলে ০ রিটার্ন করবে। যেমন :  
`printf("%d",3-1);` এই স্টেটমেন্টটি ১ প্রিন্ট করবে, কারণ '3-1' এই এক্সপ্রেশনটি সত্য। আর জটিল এক্সপ্রেশনের ক্ষেত্রে যেহেতু অ্যারিমেটিক অপারেটরের প্রিসিডেন্স বিশেষণ অপারেটরের আগে থাকে, তাই প্রথমে অ্যারিমেটিক অপারেটরের কাজ এবং পরে বিশেষণ অপারেটরের কাজ সম্পাদন হবে।

**লজিক্যাল অপারেটর :** সি-তে তিন ধরনের লজিক্যাল অপারেটর আছে। যেমন !!(not), &&(and), ||(or) এখানে ! একটি ইউনরি অপারেটর এবং এটি দিচ্ছে কোনো এক্সপ্রেশন তৈরি করা হলে তা মান হিসেবে ১ (সত্য) অথবা ০ (মিথ্যা) রিটার্ন করবে। কোনো একটি সত্য এক্সপ্রেশনের আগে

যদি ! অপারেটর ব্যবহার করা হয় তাহলে এক্সপ্রেশনটি মিথ্যা হয়ে যাবে। বাকি দুটি অপারেটর বহির্নরি অপারেটর। সুতরাং এই অপারেটর দুটি ব্যবহার করতে দুটি করে ভাটা প্রয়োজন। ধরি, a এবং b দুটি ভেরিয়েবল। যদি a এবং b উভয়ের মান সত্য (০ ছাড়া যেকোনো মান) হয়, তদু তাহলেই a&&b এই এক্সপ্রেশনের মান সত্য হবে, অন্যথা মিথ্যা হবে। আর a এবং b-এর যেকোনো একটির মান সত্য (০ ছাড়া যেকোনো মান) হলেই a|b এই এক্সপ্রেশনের মান সত্য হবে।

&& এবং || অপারেটর ব্যবহারের সময় একটি বিশেষ অবস্থার সৃষ্টি হতে পারে, যাকে বলে 'শর্টসার্কিট ইভ্যালুয়েশন/সোর্টেশন'। যেমন : a=3; b=0; c=4; তিনটি ভেরিয়েবল ডিক্লেয়ার করা হলে। এখন যদি `printf("%d",(a&&b&&c));` স্টেটমেন্টটি লেখা হয়, তাহলে তা ০ প্রিন্ট করবে। কিন্তু



এখানে লক্ষ্যীয় a এবং b অপারেটরের মধ্যে যখন && করা হচ্ছে তখনই পুরো এক্সপ্রেশনের মান ০ হয়ে যাচ্ছে অর্থাৎ এক্সপ্রেশনটি মিথ্যা হয়ে যাচ্ছে। তাই প্রোগ্রাম আর c ভেরিয়েবল নিজে কোনো কাজই করবে না। একেই বলে শর্টসার্কিট ইভ্যালুয়েশন/সোর্টেশন। অর্থাৎ && অপারেটর কোনো এক্সপ্রেশনের মাঝে ০ পেলেই শর্টসার্কিট ইভ্যালুয়েট করে আর || অপারেটর কোনো এক্সপ্রেশনের মাঝে ১ পেলেই শর্ট সার্কিট ইভ্যালুয়েট করে।

অ্যাসাইনমেন্ট অপারেটর : অ্যাসাইনমেন্ট

অপারেটরের মধ্যে বিটওয়াইজ অপারেটর ছাড়া বাকি অপারেটরগুলোর মধ্য থেকে দুইটি অপারেটরের কাজ দেখানো হবে।

++ অপারেটরকে ইনক্রিমেন্ট বলা হয়। কোনো ভেরিয়েবলের সাথে ইনক্রিমেন্ট ব্যবহার করলে তার মান ১ বেড়ে যায়। তবে ইনক্রিমেন্ট দুই রকম আছে। প্রি-ইনক্রিমেন্ট এবং পোস্ট-ইনক্রিমেন্ট। যেমন : a=1;

`printf("%d",++a); printf("%d",a++);`  
 এখানে প্রথমে প্রি-ইনক্রিমেন্ট এবং পরে পোস্ট-ইনক্রিমেন্টের উদাহরণ দেয়া হয়েছে। প্রথমে a-এর মান বেড়ে ২ হবে, তারপরে তা প্রিন্ট হবে, পরেরবার a-এর অ্যাসাইনমেন্ট হবে, তারপরে তার অ্যাসাইনমেন্ট ৩ হবে। অর্থাৎ এই তিনটি স্টেটমেন্টের অউটপুট হবে ২২ (প্রথমে ২, তারপরে আবার ২ প্রিন্ট করবে)। ডিক্রিমেন্ট অপারেটরের (-) ফেজের একইরকম নিয়ম প্রযোজ্য।

+ = এই অপারেটরের নাম ইনক্রিমেন্ট অ্যান্ড অ্যাসাইন। a=3; a+=2; এই দুটি স্টেটমেন্টের মানে হলে প্রথমে a ভেরিয়েবলের মান ৩ নির্ধারণ হবে, তারপরে a-এর মান ২ বেড়ে যাবে অর্থাৎ ৫ হবে। দ্বিতীয় স্টেটমেন্টকে a=a+2; এভাবেও লেখা যায়। অন্যান্য অপারেটরের ক্ষেত্রেও একই নিয়ম প্রযোজ্য।

ফিডব্যাক : wahid\_cseaut@yahoo.com