



সব ল্যঙ্গুয়েজেরই কিছু না কিছু প্রাথমিক নিয়মকানুন থাকে, যেগুলো না জানলে ল্যঙ্গুয়েজটি ব্যবহার করা যায় না। প্রোগ্রামিংয়ে যে ল্যঙ্গুয়েজই ব্যবহার করা হোক না কেনো, বিভিন্ন অ্যাডভান্সড স্টাইল ব্যবহার করার আগে খেয়াল করতে হবে ল্যঙ্গুয়েজটির একদম প্রিলিমিনারি নিয়মকানুনগুলো কোডার ঠিকভাবে জানেন কিনা। সি ল্যঙ্গুয়েজের খুব বেসিক গ্রামারের মাঝে ভেরিয়েবল, অপারেটর, স্টেটমেন্ট ইত্যাদি বিবেচনা করা হয়। এ লেখায় দেখানো হয়েছে কিভাবে ভেরিয়েবল এবং অপারেটর আলাদাভাবে ও একসাথে সঠিক নিয়মে ব্যবহার করা যায়।

ভেরিয়েবল সম্পর্কে প্রাথমিক ধারণা

প্রোগ্রামিংয়ে কোনো সংখ্যা নিয়ে কাজ করার জন্য প্রথমে সেই সংখ্যাকে কমপিউটারের মেমরিতে রাখতে হয়। অর্থাৎ কমপিউটারের হিসাব করা পদ্ধতিটা এমন- প্রথমে মেমরিতে ১ সংখ্যাটি রাখা হলো, তারপর মেমরির আরেক জায়গায় ২ সংখ্যাটি রাখা হলো। এবার সংখ্যা দুটির যোগফল বের করে সেই যোগফল মেমরির আরেক জায়গায় রাখা হলো। এবার ইউজার চাইলে যোগফলটি ব্যবহার করতে পারেন। কিন্তু মেমরিতে সংখ্যাগুলো রাখা একটি জটিল প্রক্রিয়া। র‍্যাম হলো কমপিউটারের প্রধান মেমরি এবং এখানেই কমপিউটার সব ডাটা রাখে। র‍্যামে অসংখ্য মেমরি সেল থাকে এবং এ সেলগুলোই মেমরির গঠনগত একক। প্রতিটি সেলের একটি নির্দিষ্ট অ্যাড্রেস থাকে। সংখ্যাগুলো এসব নির্দিষ্ট সেলে রাখা হয়। প্রথমে প্রোগ্রামে একটি সংখ্যা ডিক্লেয়ার করে তার জন্য একটি নির্দিষ্ট অ্যাড্রেস নির্ধারণ করে দেয়া হয়। কিন্তু এই কাজটি অত্যন্ত ঝামেলাপূর্ণ, কারণ র‍্যামে লাখ লাখ মেমরি সেল থাকে। এই ঝামেলা দূর করার জন্য ব্যবহার করা হয় ভেরিয়েবল, যা হাই লেভেল ল্যঙ্গুয়েজের একটি বৈশিষ্ট্য। যেমন সি-তে যদি a, b, c নামে তিনটি ভেরিয়েবল ডিক্লেয়ার করা হয়, তাহলে প্রোগ্রাম নিজে থেকেই এই তিনটি ভেরিয়েবলের জন্য মেমরিতে আলাদা জায়গা নির্ধারণ করে দেবে। তখন ইউজারের কাজ শুধু এই ভেরিয়েবলের জন্য মান নির্ধারণ করে দেয়া। অন্যভাবে বলা যায়, কোনো ভেরিয়েবল a ডিক্লেয়ার করার মানে প্রোগ্রাম যেকোনো একটি মেমরি সেল নির্বাচন করবে এবং সেই সেলটির নাম দেবে a। এখানে ইউজারকে আর কষ্ট করে অ্যাড্রেস নিয়ে কাজ করতে হবে না।

ভেরিয়েবলের নামকরণ পদ্ধতি

ভেরিয়েবল ডিক্লেয়ার করার সময় চারটি নিয়ম মেনে চলতে হয়। ০১. কোনো ভেরিয়েবলের প্রথম অক্ষর কখনও কোনো সংখ্যা হতে পারবে না। ০২. ভেরিয়েবলের নামে underscore(_) এবং dollar sign(\$) ছাড়া অন্য কোনো বিশেষ চিহ্ন ব্যবহার করা যাবে না। ০৩. ভেরিয়েবলের নামের মাঝে কোনো ফাঁকা জায়গা

থাকতে পারবে না অর্থাৎ ভেরিয়েবলের নাম সবসময় একটি শব্দ হতে হবে। ০৪. সি-তে কোনো keyword-এর নাম ভেরিয়েবলের নাম হিসেবে ব্যবহার করা যাবে না। উদাহরণ হিসেবে integer_type, auto_key, var1 ইত্যাদি।

ডাটা টাইপ সম্পর্কে ধারণা

একটি ভেরিয়েবল কী ধরনের ডাটা নিয়ে কাজ করবে তা ঠিক করে দেয়া হলো ডাটা টাইপের কাজ। যেমন- কোনো ভেরিয়েবলের ডাটা হতে পারে কোনো পূর্ণ সংখ্যা (যেমন- ৪২, ৫৩ ইত্যাদি), কোনো ভগ্নাংশ (যেমন- ৮.১৪ ইত্যাদি) অথবা কোনো অক্ষর (যেমন- a, b, c ইত্যাদি)। সি-তে প্রধানত চার ধরনের ডাটা টাইপ থাকে। এগুলো হলো character (লিখতে হয় char), integer (লিখতে হয় int), float, double। এখন দেখা যাক কোন ডাটা টাইপের জন্য মেমরিতে কী পরিমাণ জায়গা নির্ধারণ করা হয়। একটি int টাইপ ভেরিয়েবল মোট ২ বাইট (১৬ বিট) জায়গা নেয়। float-এর জন্য ৪ বাইট, double-এর জন্য ৮ বাইট জায়গা নেয়া হয় এবং এরা ভগ্নাংশ রাখতে পারে।

ডাবল কোটেশনের ভেতরে %d ব্যবহার করা হয়েছে এবং এর মাধ্যমেই কম্পাইলার বুঝতে পারে যে কোন টাইপের ডাটা ইনপুট দেয়া হচ্ছে। char-এর জন্য %, int-এর জন্য %d, float, double-এর জন্য %f ব্যবহার করা হয়। আর a ভেরিয়েবলের আগে যে & ব্যবহার করা হয়েছে এর নাম অ্যাড্রেস অপারেটর। এর মাধ্যমে a ভেরিয়েবলটির জন্য মেমরিতে যে অ্যাড্রেস নির্ধারণ করা হয়েছে তা কম্পাইলারকে জানানো হয়।

ভেরিয়েবল আউটপুট/প্রিন্ট করার নিয়ম

কোনো কিছু প্রিন্ট করার জন্য ব্যবহার করা হয় printf();। কোনো ভেরিয়েবলকে প্রিন্ট করতে হলে ডাবল কোটেশনের ভেতরে f_s রাখতে হয় এবং কোটেশনের বাইরে কমা দিয়ে ভেরিয়েবলের নাম লিখতে হয়। যেমন- printf(“%d%d\n”,a,b); এই স্টেটমেন্ট দিয়ে দুটি ভেরিয়েবলের প্রিন্ট করার কমা দেয়া হচ্ছে। ডাবল কোটেশনের ভেতরে দুটো ভেরিয়েবলের জন্য f_s ব্যবহার করা হয়েছে

সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

ভেরিয়েবল ডিক্লেয়ার নিয়ম

কোনো ভেরিয়েবল ডিক্লেয়ার করার সাধারণ নিয়ম হলো data type_name;। যেমন- int id_no; float mark; ইত্যাদি। তবে একই ধরনের অনেক ভেরিয়েবল ডিক্লেয়ার করতে হলে বারবার ডাটা টাইপ লিখতে হয় না। যেমন- int id, batch, code;। ভেরিয়েবল ডিক্লেয়ার করার সময়ই তার মান নির্ধারণ করে দেয়া যায়, যেমন- int id=248;। একই নাম একাধিক ভেরিয়েবলের জন্য ব্যবহার করলে কম্পাইলার এরর দেখাবে। আর প্রোগ্রামে কোনো ভেরিয়েবল ব্যবহার করার জন্য অবশ্যই তাকে আগে ডিক্লেয়ার করে নিতে হবে। তা না হলে কম্পাইলার এরর দেখাবে। সবসময় কোনো প্রোগ্রামের শুরুতে ভেরিয়েবল ডিক্লেয়ার করা উচিত, অন্যথায় মাঝে মাঝে এরর দেখাতে পারে।

ভেরিয়েবল ইনপুট নেয়ার নিয়ম

scanf() ফাংশন দিয়ে কোনো ডাটা ইনপুট নেয়া হয়। কিন্তু বিভিন্ন টাইপের ভেরিয়েবলের জন্য ভিন্ন ধরনের কোড লিখতে হয়। scanf() ফাংশনের ভেতরে একটা format_specifier দিতে হয়, যাতে কম্পাইলার বুঝতে পারে যে ইউজার কোন ধরনের ডাটা ইনপুট দিচ্ছে। একটি int টাইপের ভেরিয়েবলের ইনপুট নেয়ার স্টেটমেন্ট হলো scanf(“%d",&a); এখানে

এবং কোটেশনের বাইরে কমা দিয়ে ভেরিয়েবল দুটির নাম লেখা হয়েছে। এখানে লক্ষণীয়, প্রথম f_sটি প্রথম ভেরিয়েবলের জন্য এবং দ্বিতীয়টি পরের ভেরিয়েবলের জন্য কাজ করবে। আর \n দিয়ে নতুন লাইন বোঝায় অর্থাৎ যখন এটি প্রিন্ট করা হবে তখন কার্সর নিচের লাইনে চলে যাবে।

অপারেটর সম্পর্কে প্রাথমিক ধারণা

অপারেটর বলতে এমন 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 ল্যঙ্গুয়েজেও বিভিন্ন হিসাব-নিকাশের জন্য অনেক ধরনের 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। কাজের ভিত্তিতে সি-এর সব অপারেটরকে মোট ৬ ভাগে ভাগ করা যায়। যেমন :

- অ্যারেথমেটিক অপারেটর : +, -, *, /, %
- রিলেশনাল অপারেটর : >, <, >=, <=, =, ==
- লজিক্যাল অপারেটর : !, &&, ||
- বিটওয়াইজ অপারেটর : ~, &, |, ^, <<, >> ▶

- অ্যাসাইনমেন্ট অপারেটর : =, ++, --, +=, -=, *=, /=, %=, |=, &=, ^=, <<=, >>=
- সিলেকশন অপারেটর : ?:
অপারেটর যেসব ডাটা নিয়ে কাজ করে তাদের অপারেটর বলে। অপারেটরের ওপর ভিত্তি করে অপারেটরকে তিন ভাগে ভাগ করা যায়। যেমন :
• **ইউনারি অপারেটর** : যেসব অপারেটর শুধু একটি ভেরিয়েবল নিয়ে কাজ করে তাদের ইউনারি অপারেটর বলে। যেমন- প্রোগ্রামে যদি কোনো ভেরিয়েবলের 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 ডাটা নিয়ে কাজ করে তাদের বাইনারি অপারেটর বলে। যেমন- $a=2-5$ ।
- **টারনারি অপারেটর** : সি-তে একটি মাত্র অপারেটর আছে, যা তিনটি ডাটা নিয়ে কাজ করে। সিলেকশন অপারেটরকেই টারনারি অপারেটর বলে। অপারেটরটি হলো ? : এবং এটি if-else স্টেটমেন্টের সংক্ষিপ্ত রূপ হিসেবে কাজ করে। এই অপারেটরের সিনট্যাক্স হলো (condition) ? true : false । এখানে কন্ডিশনের অংশটুকু সত্য হলে true অংশ কাজ করবে, অন্যথায় false অংশ কাজ করবে।

অপারেটর প্রিসিডেন্স এবং অ্যাসোসিয়েটিভিটি

প্রোগ্রামে যে সবসময় সাধারণ এক্সপ্রেশন থাকবে তা নয়, জটিল এক্সপ্রেশনও থাকতে পারে। যেমন : $x=5/4-6*2+81\%9*(-2*-2)+(42+53)$ একটি জটিল এক্সপ্রেশন। এখানে কোন অপারেটরের আগে কে কাজ করবে সে বিষয়টিকে বলা হয় অপারেটর প্রিসিডেন্স, আর একই ধরনের অপারেটরগুলো এক্সপ্রেশনের ডান 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 প্রিসিডেন্স যোগ/বিয়োগের থেকে আগে। নিচে এই অপারেটরগুলো নিয়ে বিস্তারিত আলোচনা করা হয়েছে।

অ্যারেথমেটিক অপারেটর

+, - এই দুটি অপারেটর ইউনারি এবং বাইনারি দু'ভাবেই ব্যবহার করা সম্ভব। আর $\sqrt{\quad}$, /, % এই অপারেটর তিনটি বাইনারি অপারেটর হিসেবে ব্যবহার করা হয়। এখানে % দিয়ে ভাগশেষ বোঝান হয়। যেমন- $a=18\%6$, $b=3\%2$, $c=42\%53$, এখানে $a=0$, $b=1$, $c=42$ হবে কেননা ১৮-কে ৬ দিয়ে ভাগ করলে ভাগশেষ হয় ০, ৩-কে ২ দিয়ে ভাগ করলে ভাগশেষ হয় ১, ৪২-কে ৫৩ দিয়ে ভাগ করলে ভাগশেষ হবে ৪২, কারণ এখানে ভাগফলে কোনো পূর্ণ সংখ্যা নেই।

রিলেশনাল অপারেটর

>, <, >=(greater than or equal), <=(less than or equal) এই অপারেটরগুলো সবার কাছেই পরিচিত। কিন্তু এখানে নতুন দুটি অপারেটর আছে !=(not equal), ==(equal) এই অপারেটরগুলো সাধারণ গণিতে দেখা যায় না। যদি প্রোগ্রামে এমন একটি এক্সপ্রেশন থাকে $a=b$ তাহলে তা বোঝাবে যে '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 কিন্তু 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 কিনা তা চেক করবে।

এখানে আরেকটি কথা বলে রাখা ভালো, সি-তে লজিক্যাল এক্সপ্রেশনও একটি মান রিটার্ন করে। কোনো রিলেশনাল এক্সপ্রেশন যদি সত্য হয় তাহলে তা ১ রিটার্ন করবে, আর যদি মিথ্যা হয় তাহলে ০ রিটার্ন করবে। যেমন- $\text{printf}("%d",3>1)$; এই স্টেটমেন্টটি ১ প্রিন্ট করবে, কারণ '৩>১' এই এক্সপ্রেশনটি সত্য। আর জটিল এক্সপ্রেশনের ক্ষেত্রে যেহেতু অ্যারেথমেটিক অপারেটরের প্রিসিডেন্স রিলেশনাল অপারেটরের থেকে আগে, তাই প্রথমে অ্যারেথমেটিক অপারেটরের কাজ এবং পরে রিলেশনাল অপারেটরের কাজ সম্পাদন হবে।

লজিক্যাল অপারেটর

সি-তে তিন ধরনের লজিক্যাল অপারেটর আছে। যেমন- !(not), &&(and), |(or)। এখানে ! একটি ইউনারি অপারেটর এবং এটি দিয়ে কোনো এক্সপ্রেশন তৈরি করা হলে তার 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 বাইনারি অপারেটর। সূত্রাং এই অপারেটর দুটি ব্যবহার করতে দুটি করে ডাটা প্রয়োজন। ধরুন a এবং b দুটি ভেরিয়েবল। যদি a এবং b উভয়ের মান সত্য (০ ছাড়া যেকোনো মান) হয়, শুধু তাহলেই $a\&\&b$ এই এক্সপ্রেশনের মান সত্য হবে, অন্যথায় 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 'শর্ট সার্কিট ইন্ডালুয়েন্স/নোটেশন'। যেমন- $a=3$; $b=0$; $c=4$; তিনটি ভেরিয়েবল ডিক্লেয়ার করা হলো। এখন যদি $\text{printf}("%d",(a\&\&b\&\&c))$; স্টেটমেন্টটি লেখা হয় তাহলে তা ০ প্রিন্ট করবে। কিন্তু এখানে লক্ষণীয়, দুটি অপারেটরের মধ্যে যখন && করা হচ্ছে তখনই পুরো এক্সপ্রেশনের মান ০ হয়ে যাচ্ছে অর্থাৎ এক্সপ্রেশনটি মিথ্যা হয়ে

যাচ্ছে। তাই প্রোগ্রাম আর c ভেরিয়েবল নিয়ে কোনো কাজই করবে না। একেই বলে শর্ট সার্কিট ইন্ডালুয়েন্স/নোটেশন। অর্থাৎ && অপারেটর কোনো এক্সপ্রেশনের মাঝে ০ পেলেই শর্ট সার্কিট ইন্ডালুয়েন্স করে আর || অপারেটর কোনো এক্সপ্রেশনের মাঝে ১ পেলেই শর্ট সার্কিট ইন্ডালুয়েন্স করে।

অ্যাসাইনমেন্ট অপারেটর

++ অপারেটরকে ইনক্রিমেন্ট বলা হয়। কোনো ভেরিয়েবলের সাথে ইনক্রিমেন্ট ব্যবহার করলে তার মান ১ বেড়ে যায়। তবে ইনক্রিমেন্ট দুই ধরনের। প্রি-ইনক্রিমেন্ট ও পোস্ট-ইনক্রিমেন্ট। যেমন- $a=1$; $\text{printf}("%d",++a)$; $\text{printf}("%d",a++)$; এখানে প্রথমে প্রি-ইনক্রিমেন্ট এবং পরে পোস্ট-ইনক্রিমেন্টের উদাহরণ দেয়া হয়েছে। প্রথমে a-এর মান বেড়ে ২ হবে, তারপর তা প্রিন্ট হবে। পরেরবার a-এর ভ্যালু প্রিন্ট হবে, তারপর তার ভ্যালু বেড়ে ৩ হবে। অর্থাৎ এই তিনটি স্টেটমেন্টের আউটপুট হবে ২২ (প্রথমে ২, তারপর আবার ২ প্রিন্ট করবে)। ডিক্রিমেন্ট অপারেটরের (--) ক্ষেত্রেও একই ধরনের নিয়ম প্রযোজ্য।

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

ভেরিয়েবল এবং অপারেটর সঠিক নিয়মে ব্যবহার করে যেকোনো হিসাব-নিকাশের কাজ করা সম্ভব। তাছাড়া এগুলো বেসিক গ্রামার। তাই এ দুটি সম্পর্কে ভালোভাবে জানা প্রয়োজন [✉](mailto:wahid_cseaut@yahoo.com)

ফিডব্যাক : wahid_cseaut@yahoo.com

কার্যকাজ বিভাগে লিখুন

কার্যকাজ বিভাগের জন্য প্রোগ্রাম ও সফটওয়্যার টিপস বা টুকটাকি লিখে পাঠান। লেখা এক কলামের মধ্যে হলে ভালো হয়। সফট কপি সহ প্রোগ্রামের সোর্স কোডের হার্ড কপি প্রতি মাসের ২০ তারিখের মধ্যে পাঠাতে হবে।

সেরা ৩টি প্রোগ্রাম/টিপসের লেখককে যথাক্রমে ১,০০০, ৮৫০ ও ৭০০ টাকা পুরস্কার দেয়া হয়। সেরা ৩ টিপস ছাড়াও মানসম্মত প্রোগ্রাম/টিপস ছাপা হলে তার জন্য প্রচলিত হারে সম্মানী দেয়া হয়। প্রোগ্রাম/টিপসের লেখকদের নাম কমপিউটার জগৎ-এর বিসিএস কমপিউটার সিটি অফিস থেকেও জানা যাবে। পুরস্কার কমপিউটার জগৎ-এর বিসিএস কমপিউটার সিটি অফিস থেকে সংগ্রহ করতে হবে। সংগ্রহের সময় অবশ্যই পরিচয়পত্র দেখাতে হবে এবং পুরস্কার চলতি মাসের ৩০ তারিখের মধ্যে সংগ্রহ করতে হবে।