সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

একজন প্রোগ্রামারের বেসিক শুরু হয় সি ল্যাঙ্গুয়েজ দিয়ে। এটিকে সব ল্যাঙ্গুয়েজের ভিত্তি ধরা হয়। কিন্তু সি ও সি++ দুটি ভিন্ন ল্যাঙ্গুয়েজ। মূলত সি++ হলো সি ল্যাঙ্গুয়েজের এক্সটেনশন। আজকে সি ও সি++ এর মাঝে পার্থক্য নিয়ে আলোচনা করা হয়েছে।

সি++ একটি একক প্রোগ্রামিং ল্যাঙ্গুয়েজ। এর মাধ্যমেও সি-এর মতো একটি পূর্ণ প্রোগ্রাম লেখা যায়। এখানেও সি-এর মতো বিভিন্ন ভেরিয়েবল, অ্যারে, অপারেটর, ফাংশন, পয়েন্টার, স্ট্রাকচার ইত্যাদি ব্যবহার করতে হয়। এ ছাড়া এতে অবজেক্ট ওরিয়েন্টেড পদ্ধতিতে প্রোগ্রাম লেখার জন্য আরও কিছু বাড়তি উপাদান আছে, যা সি-তে নেই। ১৯৮০ সালে যুক্তরাষ্ট্রের বেল ল্যাবরেটরিতে ইয়ার্ন স্ট্রুস্ট্রাপ এই ল্যাঙ্গুয়েজ ডেভেলপ করেন। সি-তে যে নিয়মে ভেরিয়েবল, লুপ, ফাংশন পয়েন্টার ইত্যাদি ব্যবহার করা হয়, সি++ এ-ও একই উপায়ে ব্যবহার করা যায়। তবে সি++ এর অনেক বাড়তি সুবিধার মধ্যে একটি হলো এখানে ভেরিয়েবলকে প্রোগ্রামের যেকোনো জায়গায় ব্যবহার করা যায়, যা সি-তে সম্ভব নয়। সি-তে প্রোগ্রামের শুরুতে ভেরিয়েবল ডিক্লেয়ার না করলে কম্পাইলার সেই ভেরিয়েবল খুঁজে পেত না, যার কারণে কম্পাইলের সময় এরর দেখাত।

আসলে সি++ হলো প্রোগ্রামারদের অতিপরিচিত সি ল্যাঙ্গুয়েজের একটি আপগ্রেডেড ভার্সন। আর তাই বেসিক নিয়মগুলো এখানে আর নতুন করে শেখার দরকার পরে না। যেমন− ইনহেরিট্যান্স, পলিমরফিজম, ওভারলোডিং ইত্যাদি কনসেপ্ট এখানেও একই। তবে এই ফিচারগুলো বেসিক সি ল্যাঙ্গুয়েজে ছিল না, কারণ, এগুলো হলো অবজেক্ট ওরিয়েন্টেড ল্যাঙ্গুয়েজের ফিচার। সি কোনো অবজেক্ট ওরিয়েন্টেড ল্যাঙ্গুয়েজ নয়।

সোর্স ফাইল এক্সটেনশন : সি-এর সোর্স ফাইলকে .c দিয়ে সেভ করতে হয়, কিন্তু সি++ এর জন্য তা হলো .cpp। এভাবে সেভ না করলে তা কম্পাইলই করবে না।

সি++ এর ইন্ট্রিন্সিক ডাটা টাইপ : সি++ এর নিজস্ব ডাটা টাইপের মাঝে রয়েছে char, int, float, double, void, wchar_t, এবং bool, অর্থাৎ অতিরিক্ত দুটি ডাটা টাইপের সুবিধা।

বুল ডাটা টাইপের মান মাত্র দুটি, হয় সত্যি না হলে মিথ্যা। অর্থাৎ হয় ০ না হলে ১। মূলত কোনো লজিকাল এক্সপ্রেশনের ফলাফল রাখার জন্য এই ডাটা টাইপের ভেরিয়েবল ব্যবহার করা হয়। যদি এক্সপ্রেশনের মান সত্যি হয়, তাহলে ভেরিয়েবলের মান ১ নির্ধারিত হবে, অন্যথায় ০। এখন প্রশ্ন আসতে পারে, যদি ০ আর ১ দিয়েই কাজ হয়ে যায়, তাহলে বুল বা বুলিয়ান ভেরিয়েবলের দরকার কী? আসলে সব জায়গায় সংখ্যা ব্যবহার করা যায় না। এমন অনেক ফাংশন বা লজিকের প্যারামিটার আছে, যেখানে সরাসরি সংখ্যা ব্যবহার করলে সমস্যা হয়। তা ছাড়া কোনো কন্ডিশনের মান ০ বা ১-এর মাঝে সীমাবদ্ধ রাখার জন্যও বুলিয়ান ব্যবহার করা হয়। আবার একটি বুলিয়ান ভেরিয়েবলের মান সরাসরি সংখ্যা দিয়ে ডিক্লেয়ার করা যায় না। এর জন্য true অথবা false কিওয়ার্ড ব্যবহার করতে হয়। এই দুটি কিওয়ার্ডও সি++ এর নতুন সংযোজন। আবার অনেক পুরনো কম্পাইলার যেমন টার্বো সি++ ৩.০-তে বুলিয়ান ভেরিয়েবলের সুবিধা রাখা হয়নি। তাই সেখানে বুলিয়ান ব্যবহার করলে এরর দেখাবে। তবে আধুনিক সব কম্পাইলারেই বুলিয়ানের সুবিধা রাখা হয়েছে এবং একজন প্রোগ্রামারেরও উচিত পুরনো জিনিস ব্যবহার না করা।

সি++ এ ভেরিয়েবল ডিক্লারেশন : সি++ এর ক্ষেত্রে ভেরিয়েবল ডিক্লারেশনের ক্ষেত্রে বাড়তি সুবিধা হলো এখানে প্রোগ্রামের যেকোনো জায়গায় ভেরিয়েবল ডিক্লেয়ার করা যায়। কিন্তু সি-তে প্রথমেই সব ভেরিয়েবল ডিক্লেয়ার করতে হতো। প্রথমদিকে তা না করলে এরর দিত, কিন্তু টার্বো সি-এর পরের ভার্সনে ওয়ার্নিং দেখাত, অর্থাৎ সেটি এরর না তবে প্রোগ্রাম রান করার সময় যেকোনো ঝামেলা হতে পারে। কিন্তু সি++ এ এ ধরনের কোনো সমস্যা নেই। আবার কোনো ফর লুপের ডিক্লারেশনের ভেতরে কন্ডিশনের আগে অর্থাৎ প্রথম অংশে যদি কোনো ভেরিয়েবল ডিক্লেয়ার করা হয়, তাহলে তা প্রোগ্রামের পরে অন্যান্য জায়গায়ও ব্যবহার করা যাবে এবং ওই নামে অন্য কোনো ভেরিয়েবল ডিক্লেয়ার করা যাবে না। কিন্তু সি-এর ক্ষেত্রে লুপের বাইরে ওই ভেরিয়েবল ব্যবহার করলে কম্পাইলার খুঁজে পাবে না।

সি++ এ ভেরিয়েবলের মান নির্ধারণ : সি-তে যেভাবে অ্যাসাইনমেন্ট অপারেটরের মাধ্যমে কোনো ভেরিয়েবলের মান নির্ধারণ করা হয়, সি++ এও একই নিয়মে তা করা যায়। তবে কোনো ভেরিয়েবলের মান নির্ধারণের সময় সি++ এর নতুন সংযোজন হলো বেজ টাইপ কনস্ট্রাক্টর। এ ক্ষেত্রে কোনো ভেরিয়েবল ডিক্লেয়ার করার সময় ভেরিয়েবলের মান অ্যাসাইনমেন্ট অপারেটরের পরিবর্তে এভাবে নির্ধারণ করা যায় : int x(5), float y(3.0)।

সি++ এ লোকাল ও গ্লোবাল ভেরিয়েবল : প্রোগ্রামে যদি একই নামে লোকাল ও গ্লোবাল ভেরিয়েবল থাকে তাহলে সি-এর মতো সি++ এও ফাংশনের মাঝে সাধারণত লোকাল ভেরিয়েবলের ডাটাই ব্যবহার হয়। তবে সি++ এর বাড়তি সংযোজন হলো লোকাল ও গ্লোবাল ভেরিয়েবলের নাম যদি একই হয়, তাহলে গ্লোবাল ভেরিয়েবল ব্যবহার করার জন্য ভেরিয়েবলের নামের আগে :: সাইন ব্যবহার করতে হবে। একে বলে স্কোপ রেজ্যুলেশন অপারেটর।

সি++ এ কনস্ট্যান্ট ভেরিয়েবল : সি-এর মতো সি++ এও কনস্ট্যান্ট ভেরিয়েবল ব্যবহার করা যায় এবং সে ক্ষেত্রে const কিওয়ার্ড ব্যবহার করতে হয়। তবে এ ক্ষেত্রে ব্যতিক্রম হলো সি++ এ কনস্ট্যান্ট ভেরিয়েবল ডিক্লেয়ার করার সময়ই তার মান নির্ধারণ করে দিতে হবে। সি++ এ একটি কনস্ট্যান্ট ভেরিয়েবল সাধারণত ইন্টিজার হিসেবে কাজ করে। তাই একে প্রয়োজনে সাবস্ক্রিপ্ট হিসেবেও ব্যবহার করা যেতে পারে। তবে সি-তে ডিফাইন কিওয়ার্ড ব্যবহার করে কনস্ট্যান্ট ডিক্লেয়ার করতে হতো আর সি++ এ const কিওয়ার্ড ব্যবহার করতে হয়। ডিফাইন ও const-এর মাঝে একটি পার্থক্য আছে। তা হলো ডিফাইনে যে আইডেনিটফায়ার ব্যবহার করা হয় তার কোনো টাইপ থাকে না, কিন্তু const-এর থাকে। আর তাই const-কে কোনো ফাংশনের আর্গুমেন্ট হিসেবে পাঠালে টাইপ মিস ম্যাচ এরর হওয়ার সম্ভাবনা কম থাকে।

সি++ এ কমেন্ট ব্যবহার : সি++ এ সি-এর মতো কমেন্ট ব্যবহার করা যায়। সি-তে যেমন /*...*/ এর ভেতরে যা লেখা হতো, তাই কমেন্ট হয়ে যেত। সি++ এর জন্যও একই নিয়ম প্রযোজ্য। কিন্তু এতে বাড়তি সুবিধা হলো সি++ এ সিঙ্গেল লাইন কমেন্ট করা যায়। এখানে // সাইনের পর যা কিছু লেখা হবে, তাই কমেন্ট হিসেবে গণ্য করা হবে। তবে সেটি শুধু একটি লাইন পর্যন্ত সীমাবদ্ধ থাকবে। পরের লাইনে আবার স্বাভাবিক কোড শুরু হবে। এ ক্ষেত্রে সুবিধা হলো কমেন্টের কোনো ক্লোজিং বন্ধনী দিতে হয় না।

সি++ এ টাইপ কাস্টিং : সি-তে যেভাবে টাইপ কাস্ট করা হয় সি++ এও একইভাবে করা যায়। এখানে বাড়তি সুবিধা হলো, সি-তে সাধারণত ডাটা টাইপের সাথে প্রথম ব্রাকেট ব্যবহার করে টাইপ কাস্ট করা হতো, কিন্তু সি++ এ এভাবেও কাস্ট করা যায়, আবার আগে ডাটা টাইপ লিখে পরে ভেরিয়েবলের সাথেও প্রথম ব্রাকেট ব্যবহার করা যায়।

সি++ এ ক্যারেকটার টাইপের অ্যারের মান নির্ধারণ : সি-তে কোনো ক্যারেকটার অ্যারের মান নির্ধারণ এভাবে করতে হতো : char username[5]=”Wahid”;, কিন্তু সি++ এ তা এভাবে করতে হয় : char username[]=”Wahid”;। এ ছাড়া যতগুলো এলিমেন্ট অ্যারেতে ডিক্লেয়ার করা হয় তত মানও একসাথে নির্ধারণ করা যায়।

সি ও সি++ দুটি খুবই কাছাকাছি সিনটেক্সের ল্যাঙ্গুয়েজ। তবে সি++ থেকেই আধুনিক ল্যাঙ্গুয়েজের শুরু হয়েছে বলা যায়। তাই সি++ এর বাড়তি সিনটেক্সগুলো শেখা খুব গুরুত্বপূর্ণ কজ

ফিডব্যাক : wahid_cseaust@yahoo.com