

মিড লেভেল ল্যাপটপেজ হওয়ায় সি-এর অনেক ধরনের বৈশিষ্ট্য বিদ্যমান। হাই লেভেলের মতো ভেরিয়েবল নিয়েও এখানে কাজ করা যায়। আবার লো লেভেলের মতো অ্যাক্সেস নিয়েও কাজ করা যায়। সি-তে সরাসরি অ্যাক্সেস নিয়ে কাজ করার জন্য ব্যবহার হয় পয়েন্টার। এ লেখায় পয়েন্টার কী এবং তা কিভাবে কাজ করে তাই নিয়ে বিস্তারিত আলোচনা করা হয়েছে।

পয়েন্টার হলো এক বিশেষ ধরনের ভেরিয়েবল, যা নির্দিষ্ট টাইপের ভেরিয়েবলের অ্যাক্সেস ধারণ করতে পারে। পয়েন্টার ব্যবহার করে একজন প্রোগ্রামার সরাসরি অ্যাক্সেস নিয়ে কাজ করতে পারেন। তবে সরাসরি অ্যাক্সেস নিয়ে কাজ করা হলো লো লেভেল ল্যাপটপেজের বৈশিষ্ট্য। সি-তে একই সাথে হাই লেভেল এবং লো লেভেল ল্যাপটপেজের বৈশিষ্ট্য উপস্থিত বলে একে মিড লেভেল ল্যাপটপেজ বলা হয়।

সব প্রোগ্রামই কিছু ডাটা এবং ইনস্ট্রাকশনের সমষ্টি। সি-তে সাধারণত ডাটা নিয়ে কাজ করার জন্য ভেরিয়েবল ডিক্লেয়ার করা হয়। অনেক ল্যাপটপেজ আছে, যেখানে ডাটা ব্যবহার করার জন্য ভেরিয়েবল ব্যবহার হয় না। তবে যে ল্যাপটপেজই হোক না কেনো, প্রোগ্রাম চলার সময় প্রতিটি ভেরিয়েবলের জন্যই (অথবা যেখানে ডাটা রাখা হয়) মেমরিতে নির্দিষ্ট জায়গা দখল করা হয়। আবার কোনো প্রোগ্রাম চলার সময় প্রথমে তা মেমরিতে লোড হয়। তারপর প্রসেসর মেমরি থেকে প্রয়োজনানুসারে ইনস্ট্রাকশন নিয়ে কাজ শুরু করে। তাই কোনো প্রোগ্রাম বানাতে হলে একজন প্রোগ্রামারকে মেমরি নিয়ে চিন্তা করতে হয়। প্রোগ্রামের ডাটাকে পরিমিতভাবে ব্যবহার করলে মেমরিও দক্ষভাবে ব্যবহার হয়। ফলে প্রোগ্রামের গতি বাড়ে এবং রান টাইম কমে। অ্যারে নিয়ে আলোচনা করার সময় দেখানো হয়েছে কিভাবে অ্যারের ব্যবহারের ফলে প্রোগ্রামের জটিলতা কমানো সম্ভব। সেই সাথে দেখানো হয়েছে পয়েন্টার ব্যবহার করেও কিভাবে জটিলতা কমানো যায়।

পয়েন্টার

পয়েন্টার হলো একটি বিশেষ ভেরিয়েবল, কিন্তু এটি কোনো সাধারণ মান ধারণ করতে পারে না। এটি শুধু অপর ভেরিয়েবলের অ্যাক্সেস ধারণ করতে পারে। একটি প্রোগ্রামের জন্য অ্যাক্সেসই মূল বিষয়। প্রতিটি ভেরিয়েবলেরই একটি করে অ্যাক্সেস থাকে। প্রোগ্রাম ওই ভেরিয়েবলগুলোকে তাদের নামে নয় বরং তাদের অ্যাক্সেস দিয়ে চেনে। ওই অ্যাক্সেসে কোনো কিছু পরিবর্তন করলে সংশ্লিষ্ট ভেরিয়েবলেও সেই পরিবর্তন দেখা যাবে। অর্থাৎ কোনো ভেরিয়েবলের যে অ্যাক্সেস আছে, সে অ্যাক্সেসের মানকে মুছে দেয়, ফলে ভেরিয়েবলের মানও ডিলিট হয়ে যাবে। আবার কোনো অ্যাক্সেসে নতুন কোনো মান অ্যাসাইন করা হলো ওই অ্যাক্সেসের যে ভেরিয়েবল আছে তার মানও পরিবর্তন হয়ে যাবে।

ভেরিয়েবলের অ্যাক্সেস

সি-তে ভেরিয়েবল নিয়ে কাজ করার জন্য অ্যাক্সেস অপারেটর ব্যবহার করা হয় এবং একে

‘&’ দিয়ে প্রকাশ করা হয়। এই অপারেটরটি শুধু ভেরিয়েবলের সাথে ব্যবহার করা যায়। যেমন :
 intx বলতে একটি ভেরিয়েবল বোঝাবে, কিন্তু &x দিয়ে ওই ভেরিয়েবলের অ্যাক্সেস বোঝাবে। এমনকি সি-তেও মাঝে মাঝে ভেরিয়েবলের অ্যাক্সেস উল্লেখ করে দিতে হয়। এ কারণেই কোনো ইনপুট নেয়ার সময় scanf(“%d”,&x) ফাংশনের ভেতরে ব্যবহার করা হয়। অর্থাৎ এখানে প্রোগ্রামকে বলে দেয়া হয়, ইনপুট নিয়ে সেটা x-এর অ্যাক্সেসে রেখে দাও। একইভাবে কোনো ভেরিয়েবলের অ্যাক্সেস প্রিন্ট করা সম্ভব। সাধারণত ভেরিয়েবল প্রিন্ট করার সময় সাধারণভাবে ভেরিয়েবলের নাম লেখা হয়। কিন্তু অ্যাক্সেস প্রিন্ট করতে চাইলে printf(“%d”,&x) এভাবে লিখতে হবে। মনে রাখতে হবে, এভাবে প্রিন্টের কমান্ড লিখলে x-এর মান প্রিন্ট হবে না, বরং তার অ্যাক্সেস প্রিন্ট হবে। তবে এ অ্যাক্সেস হেক্সাডেসিমেল নাম্বারে প্রিন্ট হবে। কমপিউটার সাধারণত হেক্সাডেসিমেলেরই ইনপুট নেয় এবং আউটপুট দেয়। কিন্তু প্রসেসর যখন হিসাব করে তখন বাইনারিতে করে। আবার আউটপুট যদি ইউজার ডেসিমেল চায়, তাহলে ডেসিমেলেরই দেখানো হয়।

: প্রথম লাইনটি লেখার মানে হলো gpa নামের একটি ইন্টিজার টাইপের পয়েন্টার ডিক্লেয়ার করা হলো, যা শুধু ইন্টিজার টাইপের ভেরিয়েবলের অ্যাক্সেস ধারণ করতে পারবে। দ্বিতীয় লাইনটি বোঝাচ্ছে grade নামে একটি পয়েন্টার ভেরিয়েবল ডিক্লেয়ার করা হলো, যা শুধু ক্যারেক্টার টাইপের ভেরিয়েবলের অ্যাক্সেস ধারণ করতে পারবে। এখানে আসলে দুটি অংশ আছে। একটি হলো ডাটা টাইপের পয়েন্টার। অর্থাৎ ডিক্লেয়ারেশনের প্রথম অংশ নির্দেশ করে পয়েন্টারটি কোন ধরনের ডাটা টাইপকে পয়েন্ট করতে সক্ষম হবে। আর পরের অংশ হলো পয়েন্টারের নাম। এটি সাধারণ ভেরিয়েবলের মতো যেকোনো নাম হতে পারে।

লক্ষণীয়, পয়েন্টারে সত্যিকার অর্থে কোনো ডাটা টাইপ নেই। আমরা জানি, একেক টাইপের ডাটা একেক ধরনের মান ধারণ করতে পারে। যেমন : ক্যারেক্টার টাইপের ভেরিয়েবল সর্বোচ্চ ২৫৫ পর্যন্ত মান ধারণ করতে পারে। আবার ইন্টিজার সর্বোচ্চ ৩২৭৬৭ পর্যন্ত ধারণ করতে পার। পয়েন্টারের ক্ষেত্রে এ ধরনের কোনো বিষয় নেই যে ইন্টিজার টাইপের পয়েন্টার

সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

কোনো নাম্বার হেক্সাডেসিমেল আছে কিনা, তা চেনার সহজ উপায় হলো নাম্বার যদি হেক্সাডেসিমেল হয় তাহলে তার শেষে h থাকবে।

পয়েন্টার ভেরিয়েবল

পয়েন্টারের কাজ যদিও শেষ ধরনের, তবুও এটি একটি ভেরিয়েবল। সাধারণ ভেরিয়েবলের মতো একেও ডিক্লেয়ার করতে হয়। পয়েন্টারেরও বিভিন্ন টাইপ আছে। মনে রাখতে হবে, সাধারণ ভেরিয়েবল যত ধরনের হয় পয়েন্টারও তত ধরনের হয়। যেমন : ইন্টিজারের জন্য ইন্টিজার টাইপের পয়েন্টার, ক্যারেক্টারের জন্য ক্যারেক্টার টাইপের পয়েন্টার ইত্যাদি। পয়েন্টার ডিক্লেয়ার করার নিয়ম সাধারণ ভেরিয়েবলের মতোই, তবে একটু পার্থক্য হলো এখানে পয়েন্টার করার সিম্বল ব্যবহার করতে হয়। যেমন :

```
int *gpa;      অথবা int* gpa;
char *grade;  অথবা char* grade;
double *cgpa; অথবা double* cgpa;
```

খেয়াল করতে হবে এখানে পয়েন্টার ডিক্লেয়ার করার সময় * সাইন ব্যবহার করা হয়েছে। এই সাইনটি দিয়ে বোঝানো হয় যে ডিক্লেয়ার করা ভেরিয়েবলটি একটি পয়েন্টার ভেরিয়েবল। এটি দুইভাবে ব্যবহার করা যেতে পারে। ডিক্লেয়ারেশনের সময় ভেরিয়েবলের নামের ঠিক আগে অথবা ডাটা টাইপের ঠিক পরে এই সাইনটি দিতে হয়। তবে এই দুই স্থান ছাড়া অন্য কোথাও দিলে এরর দেখাতে পারে। যেমন

একরকম, আবার ক্যারেক্টার টাইপের পয়েন্টার আরেকরকম মান ধারণ করতে পারে। সব পয়েন্টারই একই ধরনের মান ধারণ করতে পারে। একটি ইন্টিজার টাইপের পয়েন্টার যে পর্যন্ত সংখ্যা ধারণ করতে পারে, ক্যারেক্টার টাইপের পয়েন্টারও একই সংখ্যা ধারণ করতে পারে। শুধু পার্থক্য হলো, সাধারণ ভেরিয়েবলের ক্ষেত্রে ভিন্ন টাইপ মানে হলো ভেরিয়েবলের লিমিট ভিন্ন হবে, আর পয়েন্টারের ক্ষেত্রে ভিন্ন টাইপ মানে হলো ভিন্ন টাইপের ভেরিয়েবলকে পয়েন্ট করবে। এ কারণেই পয়েন্টারের জন্য ডাটা টাইপ না বলে ডাটা টাইপের পয়েন্টার বলা উচিত। অন্যভাবে বলা যায়, প্রোগ্রামে যখন সাধারণ ভেরিয়েবল ডিক্লেয়ার করা হয়, তখন কম্পাইলারকে জানিয়ে দেয়া হয় যে ভেরিয়েবলটির মাঝে কী রকম ডাটা রাখা যাবে। আর প্রোগ্রামে যখন পয়েন্টার ডিক্লেয়ার করা হয়, তখন কম্পাইলারকে জানিয়ে দেয়া হয় যে পয়েন্টারটি কোন ধরনের ভেরিয়েবলকে পয়েন্ট করতে পারবে। এখানে এমন নয় যে একেক টাইপের পয়েন্টার একেক লিমিট পর্যন্ত সংখ্যা ধারণ করতে পারবে। কারণ, মেমরির প্রতিটি সেলের একটি নিজস্ব অ্যাক্সেস আছে। তবে মেমরির প্রতিটি সেলের অ্যাক্সেস প্রকাশ করতে সমানসংখ্যক বিটের প্রয়োজন হয়। অর্থাৎ মেমরির প্রথম সেলের অ্যাক্সেস প্রকাশ করতে যদি ১০টি বিটের প্রয়োজন হয়, তাহলে মাঝের একটি সেলের অ্যাক্সেস প্রকাশ করতেও ১০টি

বিটের প্রয়োজন হবে। আর পয়েন্টার যেহেতু মেমরি সেলের অ্যাড্রেস ধারণ করে, তাই সব পয়েন্টারই সমানসংখ্যক বিট ধারণ করতে পারে।

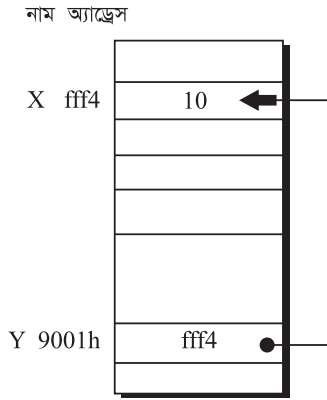
সাধারণ ভেরিয়েবল এবং পয়েন্টার দুটোর কাজ একই। তা হলো ডাটা রাখা। একজন ডাটা রাখে আর আরেকজন ডাটার অ্যাড্রেস রাখে। তাই একই স্কোপের মাঝে ভেরিয়েবল এবং পয়েন্টার দুটোর নাম একই হতে পারবে না। সাধারণ ভেরিয়েবল এবং পয়েন্টার ডিক্লেয়ার করার নিয়ম সম্পূর্ণ এক। তাই সাধারণ ভেরিয়েবলকে যেমন একসাথে ডিক্লেয়ার করা যায়, তেমনি পয়েন্টারকেও একসাথে ডিক্লেয়ার করা যায়। যেমন : `int x,y,*z;` এখানে তিনটি ইন্টিজার টাইপের ভেরিয়েবল ডিক্লেয়ার করা হয়েছে। এর প্রথম দুটি সাধারণ ভেরিয়েবল এবং তৃতীয়টি পয়েন্টার ভেরিয়েবল।

পয়েন্টারের মূল কাজ মেমরির অ্যাড্রেস ধারণ করা। কিন্তু মেমরি অ্যাড্রেসও একটি সংখ্যা। তাই সংখ্যাটি পয়েন্টারের মাঝে রাখার অর্থ মেমরির অন্য সেলে সেভ করে রাখা। এখন পয়েন্টারের ডাটা রাখার জন্য কতগুলো সেল ব্যবহার হবে তা নির্ভর করে কমপিউটারের আর্কিটেকচারের ওপর। একেক ধরনের কমপিউটারের আর্কিটেকচার একেক ধরনের হয়। তাছাড়া এটি অপারেটিং সিস্টেম, কম্পাইলার ইত্যাদির ওপরও নির্ভর করে। আবার পয়েন্টারের ডাটা রাখার জন্য যেহেতু মেমরির প্রয়োজন, তাই বলা যেতে পারে, সাধারণ ভেরিয়েবলের মতো পয়েন্টারেরও একটি নিজস্ব অ্যাড্রেস থাকে। এই অ্যাড্রেসের জন্য কত বাইট দখল করা হবে তাও নির্ভর করে আর্কিটেকচার, অপারেটিং সিস্টেম ইত্যাদির ওপর।

পয়েন্টারের মান নির্ধারণ

ডিক্লেয়ার করার পর পয়েন্টারের মান নিজে থেকে নির্ধারিত হয় না। অর্থাৎ পয়েন্টার নিজে থেকে অন্য কোনো ভেরিয়েবলকে বা অ্যাড্রেসকে পয়েন্ট করতে পারে না। এক্ষেত্রে প্রোগ্রামারকেই বলে দিতে হয় যে পয়েন্টারটি কাকে পয়েন্ট করবে। অর্থাৎ পয়েন্টারের মান অ্যাসাইন করে দিতে হবে। পয়েন্টারের মান অ্যাসাইন করার নিয়ম সাধারণ ভেরিয়েবলের মতোই, তবে

এখানে & অপারেটর ব্যবহার করতে হয়। যেমন : `int x=10, &y; y=&x;` এ দুটি লাইনের প্রথমে ইন্টিজার টাইপের একটি ভেরিয়েবল এবং একটি পয়েন্টার ডিক্লেয়ার করা হয়েছে। ভেরিয়েবলটি ডিক্লেয়ার করার সময় আবার মান অ্যাসাইন করে দেয়া হয়েছে। দ্বিতীয় লাইনে পয়েন্টারের মান অ্যাসাইন করা হয়েছে। আমরা জানি & অপারেটর সাধারণত অ্যাড্রেসের জন্য ব্যবহার হয়। তাই কোনো ভেরিয়েবলের অ্যাড্রেস বোঝাতে হলে তার আগে & অপারেটর ব্যবহার করতে হবে। তাই দ্বিতীয় লাইনে y-এর মান হিসেবে &x অর্থাৎ x-এর অ্যাড্রেস নির্ধারণ করা হয়েছে। সুতরাং x হলো একটি ইন্টিজার এবং y হলো x-এর পয়েন্টার। অন্যভাবে বলা যায় x-এর মান হলো 10, x-এর অ্যাড্রেস হলো fff4, y-এর মান হলো fff4, y-এর অ্যাড্রেস হলো 9001h। এখানে অ্যাড্রেসগুলো কল্পনা করা হয়েছে। চিত্রে বিষয়টি তুলে ধরার চেষ্টা করা হলো।



মান নির্ধারণের সময় কয়েকটি বিষয় লক্ষ করতে হবে। সাধারণ ভেরিয়েবলের মতো পয়েন্টার ভেরিয়েবলেরও ডিক্লেয়ার করার সময় মান নির্ধারণ করা যায়। যেমন : `int x,*y=&x;` তবে এক্ষেত্রে উল্লেখ্য, এভাবে পয়েন্টারের মান নির্ধারণ করার সময় পয়েন্টার অন্য যে ভেরিয়েবলকে পয়েন্ট করবে (অর্থাৎ যার অ্যাড্রেস পয়েন্টারের মাঝে থাকবে), তাকে অবশ্যই আগে ডিক্লেয়ার করতে হবে। সুতরাং `int *y=&x,x;` এভাবে লিখলে এরর দেখাবে। কারণ আমরা জানি, প্রোগ্রাম একটার পর একটা

কমান্ড এক্সিকিউট করে। তাই প্রোগ্রাম যখন y-এর মান অ্যাসাইন করবে তখন দেখবে x নামে কোনো ভেরিয়েবল নেই। কারণ x-কে পরে ডিক্লেয়ার করা হয়েছে। তাই কম্পাইলার এখানে এরর দেখাবে। পয়েন্টার ভেরিয়েবলের মান নির্ধারণের সময় এর ডাটা টাইপের দিকে খেয়াল রাখতে হয়। এক ডাটা টাইপের পয়েন্টার অন্য ডাটা টাইপের ভেরিয়েবলকে পয়েন্ট করতে পারে না। তাই পয়েন্টার এবং পয়েন্টেড ভেরিয়েবলের ডাটা টাইপ অবশ্যই এক হতে হবে। যেমন : `int x; float *y=&x;` এখানে কম্পাইলার এরর দেখাবে, কারণ ভেরিয়েবল দুটির ডাটা টাইপ এক নয়। তবে পয়েন্টারের মান নির্ধারণ করার ডাটা কাস্ট করা সম্ভব। যেমন : `int x; char *y=(char*)&x;` এখানে যদিও x এবং y-এর ডাটা টাইপ ভিন্ন, কিন্তু y-এর মান নির্ধারণ করার সময় ডাটা কাস্ট করা হয়েছে। অর্থাৎ x-এর অ্যাড্রেস ক্যারেক্টার হিসেবে y থাকবে। ডাটা কাস্ট করে পয়েন্টারের মান নির্ধারণ করলে খেয়াল রাখতে হবে যে ডাটা টাইপ কাস্ট করা হচ্ছে। তারপর অবশ্যই যেনো * সাইন দেয়া হয়। কিন্তু নিতান্ত নিরুপায় না হলে এ ধরনের ডাটা কাস্ট করে পয়েন্টারের মান নির্ধারণ করা ঠিক নয়। কারণ পয়েন্টারের জন্য এ ধরনের মান নির্ধারণ কম্পাইলার সমর্থন করে না। তাই এখানে ওয়ানিং দেখাতে পারে এবং কখনো কখনো বামেলা করতে পারে। পয়েন্টারের মান সাধারণত অপারেটরের সাহায্য ছাড়া নির্ধারণ করা যায় না। যেমন : `int x; int *y=fff4;` এখানে যদি x-এর অ্যাড্রেস fff4 হয়, তাহলে y-এর মান সরাসরি এভাবে অ্যাসাইন করা যাবে না, এরর দেখাবে।

পয়েন্টার সি ল্যান্ডুয়েজে অত্যন্ত গুরুত্বপূর্ণ ভূমিকা পালন করে। পয়েন্টার ছাড়া অন্য কোনোভাবে সি-তে অ্যাড্রেস নিয়ে কাজ করা সম্ভব নয়। বড় বড় প্রোগ্রাম বানাতে হলে অনেক সময়ই অ্যাড্রেস নিয়ে কাজ করতে হয়। আবার অনেক ধরনের ডাটা স্ট্রাকচার আছে, যেগুলো পয়েন্টারের ধারণার ওপর গঠিত। তাই পয়েন্টারের ব্যবহার ভালোভাবে শেখা খুব গুরুত্বপূর্ণ কাজ

ফিডব্যাক : wahid_cseast@yahoo.com