

সহজ ভাষায় প্রোগ্রামিং

সি/সি++

আহমদ ওয়াহিদ মাসুদ

সি দিয়ে হয়েছে এখনকার সব আধুনিক সফটওয়্যার বানানো সম্ভব নয়, তবে সব ধরনের গ্যাপুয়েঞ্জের ভিত্তি হলো সি। আসলে সি দিয়ে আধুনিক সফটওয়্যারের লজিক দাঁড় করাণো সম্ভব এবং এটিই একটি সফটওয়্যারের প্রধান অংশ।

সি-তে ফাংশন ব্যবহারের সুবিধা একদিকে যেমন প্রোগ্রামকে করে সুন্দরতর, তেমনি ইউজারের জন্য কেফিঙ করে তুলে আরও সহজ। ফাংশনের ব্যবহার নিয়ে আগের সংখ্যায় আলোচনা করা হয়েছে। এ সংখ্যায় ফাংশনের ব্যবহারের আরও গভীরে ঢোকা হয়েছে এবং দেখানো হয়েছে কীভাবে ফাংশন আসলে কাজ করে এবং তা কী কী উপায়ে ব্যবহার করা যায়।

ফাংশন প্রোটোটাইপ

অজ্ঞেই ফাংশনের প্রোটোটাইপ সম্পর্কে আলোচনা করা হয়েছে। আমরা জানি, সি-তে প্রোগ্রাম শুরু হয় মেইন ফাংশন থেকে। প্রোগ্রামে অনেক লিট্টইন ফাংশন ব্যবহার করা হয় এবং নিট্টইন ফাংশন সর্শি-উ হেডার ফাইল থেকে ইন্স্টল করা হয়। তাই অলানা কোনো কিছু করার প্রয়োজনা পড়ে না। কিন্তু ইউজার যদি নিজের নতুন ফাংশন ব্যবহার করতে চান, তাহলে প্রত্যেক ফাংশনের প্রোটোটাইপ ব্যবহার করতে হবে। ফাংশনের প্রোটোটাইপ হলো আর কিছুই নয় শুধু ফাংশনের নাম মেইন ফাংশনের কাছে লিখে দেয়া। এটি লেখার নিয়ম হলো: "রিটার্ন টাইপ ফাংশনের নাম (ফাংশনের প্যারামিটার)"।

এটি মেইন ফাংশনের আগে যেকোনো জায়গায় লিখলেই হবে, শেষে সেমিফোনন দিতে হবে। তবে এখানে একটি বিশেষ নিয়ম আছে। ইউজার ডিকারিই ফাংশনের বডি যদি মেইন ফাংশনের পরে লেখা হয়, তাহলে উল্লিখিত ফাংশনের প্রোটোটাইপ দেয়া আবশ্যিক, আর বডি যদি মেইন ফাংশনের আগেই লেখা হয় তাহলে আর প্রোটোটাইপ দিতে হবে না। অংশাটাই খেয়াল রাখতে হবে প্রোটোটাইপে দেয়া ফাংশনের রিটার্ন টাইপ, ফাংশনের নাম এবং প্যারামিটার যেন ফাংশনের বডিতে দেয়া রিটার্ন টাইপ, নাম এবং প্যারামিটার একই হয়। ফাংশনের প্যারামিটারে ব্যবহার হওয়া ডেফাইনেশনের রিটার্ন টাইপ অবশ্যই দিতে হবে। যেমন: ইউজার যদি নিট্ট ফাংশন নামে একটি ফাংশন ডেফাইন করেন এবং তার প্যারামিটারে যদি দুটি ইন্টিজার থাকে তাহলে এর প্রোটোটাইপ হবে: void newFunction(int x,int y);। ফাংশনের বডিও শুরুতেও এরকম ডেফাইনেশন দিতে হবে, শুধু পার্থক্য হলো কোনো সেমিফোনন থাকবে না। এবার ফাংশনের রিটার্ন টাইপ এবং ফাংশন দিয়ে

কীভাবে ডায়ালু রিটার্ন করা সম্ভব তা নিয়ে আলোচনা করা যাক।

রিটার্ন টাইপ মানে যেকোনো ধরনের ডাটা টাইপ অথবা যদি কিছুই রিটার্ন করার দরকার না হয় সেখানে ডায়ালু। ওপরের উদাহরণ থেকে দেখা যাচ্ছে ফাংশনটির রিটার্ন টাইপ ডায়ালু অর্থাৎ এই ফাংশনটিকে সেখানে কল করা হবে সেখানে এটি কোনো ডায়ালু রিটার্ন করবে না। ফাংশন কী রিটার্ন করবে না করবে সেটি return; স্টেটমেন্ট দিয়ে নির্ধারণ করা হয়। যদি ফাংশনকে কোনো ডায়ালু রিটার্ন করতে হয় তাহলে রিটার্ন স্টেটমেন্টের পর সেই ডায়ালু দিয়ে। আর যদি কোনো ডায়ালু রিটার্নের দরকার না হয়, তাহলে শুধু রিটার্ন লিখে সেমিফোনন দিতে হবে। অথবা কোনো রিটার্ন স্টেটমেন্ট না লিখলেও হবে, সেখানে ফাংশন সেখানে শেষ হয়ে যাবে, সেখান থেকে সে নিজেই রিটার্ন করবে। নিচে রিটার্ন নিয়ে একটি উদাহরণ দেয়া হলো। ধরা যাক, এমন একটি ফাংশন লেখার প্রয়োজনা পড়ল যেন তা একটি ইন্টিজার, একটি ক্যারেক্টার দিতে পারে এবং সেই ক্যারেক্টারের আফিঙ ডায়ালু দিয়ে (স্ট্র্যাডার্জ ডায়ালু) ওই ইন্টিজারকে গুণ করলে যে ডায়ালু পাওয়া যায়, তা রিটার্ন করে। তাহলে ফাংশনের বডি নিচের মতো হবে:

```
int newFunction(int x,char y)
{
    printf("the value is returned!!");
    return x*y;
}
```

এখানে দেখা যাচ্ছে, ফাংশনটি ইন্টিজার টাইপের ডাটা রিটার্ন করতে সক্ষম। এর দুটি প্যারামিটার আফিঙ এটি দুটি ডেরিভেট দিতে পারবে, যার একটি ইন্টিজার এবং আরেকটি ক্যারেক্টার। ডায়ালু সোয়ার পর এটি প্রদর লাইনটি প্রিন্ট করবে, তারপর নির্দিষ্ট ডায়ালু রিটার্ন করবে। ডায়ালু রিটার্ন করার অর্থ হলো যি প্রোগ্রামে কোথাও লেখা থাকে যে i=newFunction(2,A) তাহলে; এর ডায়ালু হিসেবে 1৩০ নির্ধারিত হবে, কারণ A-এর আফিঙ ডায়ালু ৬৫।

রিটার্ন শ্রেয়

সব প্রোগ্রামিয়ারের জন্য একটি সত্যি কথা হলো: কোনো কোড ইউজারের জন্য বোকা যত কর্তিই হবে তা কম্পিউটারের জন্য রান করা তত সহজ হবে, অর্থাৎ কম রিসোর্সে নেবে অথবা কম টাইম দেবে। আর কোনো কোড ইউজারের জন্য বোকা যত সহজ হবে তা কম্পিউটারের জন্য রান করা তত কর্তিই হবে অর্থাৎ সেই কোড বেশি রিসোর্স টাইমে অথবা প্রোগ্রামের রান টাইমে বেশি হবে। রিটার্ন দিতে হবে আর আডভান্সড সি

প্রোগ্রামিং শুরু। রিটার্নের মাধ্যমে প্রোগ্রামের রানটাইম অনেক কমানো সম্ভব, প্রোগ্রাম আরও এফিসিয়েন্ট করা সম্ভব এবং এতে প্রোগ্রামের জটিলতা অনেকাংশে কমে যায়। তবে রিটার্নের কোড কিছুটা কর্তি, এটি পুরোটাই কমপ্লেক্সিটি বিখ্য, তাই বুঝতে একটু কষ্ট হতে পারে, তবে রিটার্নের মাধ্যমে কোড সহজে অনেক বড় প্রোগ্রামকে যেমন আকারে অনেক ছোট করা সম্ভব, তেমনি প্রোগ্রামের রানও এতে অনেক বেড়ে যায়।

রিটার্নের বেসিক ধারণা খুব সহজ। একটি ফাংশন যখন নিজেই নিজেতে কল করে, তখন তাকে রিটার্ন বলে। তবে রিটার্ন শুধু একটি ফাংশনের মধ্যেই সীমাবদ্ধ থাকে না। একটি ফাংশনকে যদি অন্য ফাংশনের মধ্যেও ব্যবহার কল করা হয়, তাহলেও সেটি রিটার্ন হয়। খোলাসা মানে হবে, ফাংশনের এই ব্যবহার কল যেম কোনো লুপ দিতে করা না হয়। লুপ ব্যবহার করলে সেটি আর রিটার্ন করে না। রিটার্নের বেসিক উদ্দেশ্যই হলো লুপের মতো কাজ করা, কিন্তু লুপ ব্যবহার করা যাবে না। আসলে এটি লুপের বিকল্প হিসেবে কাজ করে। নিচে একটি ফাংশনের বডি দেয়া হলো। এটি মূলত কোনো নাম্বারের ফ্যাক্টোরিয়াল বের করার ফাংশন। আমরা জানি কোনো নাম্বার n-এর ফ্যাক্টোরিয়ালের সূত্র হলো n(n-1)(n-2)....

```
int fact(int n)
{
    if(n==1)
        return 1;
    else
        return n*fact(n-1);
}
```

এখানে লক্ষ্যীয়া, ফাংশনটি নিজেই নিজেতে কল করছে। প্রথম n-এর ডায়ালু যদি 1 বা তার কম হয়, তাহলে এটি 1 রিটার্ন করবে। এখন আমরা কর্তি, 1-এর ফ্যাক্টোরিয়াল 1। আর n-এর ডায়ালু যদি ৩ হয় তাহলে এটি else-এ ঢুকবে। এখানে আবার এই ফাংশনটিকে কল করা হবে এবং তার প্যারামিটার হিসেবে 2 পাঠানো হবে। অর্থাৎ এই ফাংশনে n-এর ডায়ালু 2 হলে যে ফলাফল আসবে তার সাথে 2 গুণ করে এটি রিটার্ন করবে। আবার নতুন করে যে কল করা হলো সেখানে n=2 হওয়ার কারণে প্রোগ্রাম আবার else-এ যাবে এবং আবার রিটার্ন হবে। সেখানে ফাংশনটিকে আবার কল করা হবে এবং সেখানে প্যারামিটার দেয়া হবে 1। এখানে ফাংশন 1 সহকারে রিটার্ন করবে, কারণ n=1 হলে 1 রিটার্ন করার কমান দেয়া আছে। সেই 1-এর সাথে ইন্টিজারের n অর্থাৎ 2 গুণ হবে। এভাবে 2 রিটার্ন হবে প্রথমবার যখন এই ফাংশনটিকে কল করা হয়েছে সেখানে। সেখানে রিটার্ন করা ডায়ালু হলো 2, এর সাথে বর্তমান n=3 গুণ হবে। এভাবে 6 রিটার্ন হবে মূল প্রোগ্রামে।

রিটার্ন সি প্রোগ্রামে গ্যাপুয়েঞ্জের একটি অন্যতম বৈশিষ্ট্য। এটি সি ল্যাঙ্গুয়েজকে করেছে অনেক সহজ এবং প্রকৃততর। হিকমততা রিটার্ন দি ব্যবহার করতে পারলে প্রোগ্রাম অনেক উন্নত করা সম্ভব।

ফিডব্যাক: wahid_cseuata@yahoo.com