সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

সি প্রোগ্রামিংয়ের সাথে পরিচিত সবাই কমবেশি ফাংশনের সাথেও পরিচিত। লুপিং যেমন সি প্রোগ্রামিংয়ের একটি গুরুত্বপূর্ণ বিষয়, তেমনি ফাংশনও। আসলে ফাংশন ছাড়া সি প্রোগ্রামিং করা সম্ভব নয়। ফাংশন হলো সি প্রোগ্রামিংয়ের ভিত্তি।

সি প্রোগ্রামে কোড লেখার সময় সবাই main() এই অংশটি লেখেন, যা লেখা দরকার। এটি একটি ফাংশন, যাকে মেইন ফাংশন বলা হয়। এটি ছাড়া প্রোগ্রাম রান করা সম্ভব নয়। এ ধরনের আরও অনেক ফাংশন আছে। এগুলো প্রোগ্রামের জটিলতা বহুগুণে কমিয়ে দেয়। ফাংশন সম্পর্কে ধারণা পাওয়ার জন্য একটি বইয়ের কথা চিন্তা করা যায়। একটি বই যেমন বিভিন্ন অধ্যায়ে ভাগ করা থাকে, তেমনি সি প্রোগ্রামের কোডও বিভিন্ন অংশে ভাগ করা থাকে। এই অংশগুলোকে ফাংশন বলা যেতে পারে। এদের কাজ একই এবং বিভিন্ন সময়ে একই রকম কাজ করার জন্য এদের ব্যবহার করা হয়। সি প্রোগ্রামিংয়ে printf(), scanf(), clrscr() ইত্যাদি বিভিন্ন ফাংশন আছে। এসব ফাংশনের কাজ একই। যেমন printf()-এর কাজ কোনো কিছু প্রিন্ট অর্থাৎ মনিটরে লেখানো, scanf()-এর কাজ হলো ইউজারের কাছ থেকে কিবোর্ডের কোনো ইনপুট নেয়া, clrscr()-এর কাজ হলো স্ক্রিনে, যা কিছু আছে সব মুছে ফেলা ইত্যাদি। এ কাজগুলো আসলে এত সহজ নয়, যেমন ইনপুট নেয়ার জন্য সি প্রোগ্রামিংয়ে অনেক কোড লেখার প্রয়োজন। কিন্তু scanf() লিখলেই সহজে ইনপুট নেয়া যায়। কারণ, এর জন্য প্রয়োজনীয় কোড আগে থেকে লিখে দেয়া হয়েছে। stdio.h নামের হেডার ফাইলে এই scanf() ফাংশনটি লেখা আছে। প্রোগ্রামে যখন scanf() লেখা হয়, তখন প্রোগ্রাম ওই হেডার ফাইল থেকে সংশ্লিষ্ট ফাংশনের কোডগুলো কম্পাইল করে নেয়। এভাবে ফাংশনের কাজই হলো সম্পূর্ণ প্রোগ্রামটিকে ধাপে ধাপে সম্পন্ন করা।

ফাংশন মূলত দুই ধরনের। যেমন : লাইব্রেরি ফাংশন এবং ইউজার ডিফাইন্ড ফাংশন। লাইব্রেরি ফাংশনের আরেক নাম বিল্টইন ফাংশন। হেডার ফাইলে যেসব ফাংশন বর্ণিত থাকে সেগুলো হলো লাইব্রেরি ফাংশন। এ ফাংশনগুলো আগে থেকেই লেখা আছে দেখে এরূপ নামকরণ। কম্পাইলার অনুযায়ী লাইব্রেরি ফাংশন নির্ধারিত হয়। তবে বেশিরভাগ ফাংশনই সব কম্পাইলারে অপরিবর্তিত থাকে। আবার ইন্টারনেটে অনেক এক্সটারনাল লাইব্রেরি ফাংশনও পাওয়া যায়। এগুলো ব্যবহার করে প্রোগ্রাম আরও সহজে চালানো সম্ভব। লাইব্রেরি ফাংশনের সুবিধা বোঝানোর জন্য একটি ছোট উদাহরণ দেখা যাক :

```
#include<conio.h>
void main()
{
      clrscr();
}
```

ওপরের প্রোগ্রামটির কাজ হলো স্ক্রিনে যা কিছু আছে সব মুছে দেয়া। এখানে clrscr(); একটি লাইব্রেরি ফাংশন। কিন্তু যদি লাইব্রেরি ফাংশন ব্যবহার না করা হয় তাহলে ওপরের কাজটি করার জন্য নিচের প্রোগ্রাম লিখতে হবে :

```
#include<dos.h>
void main()
{
      union REGS r;
r.h.ah=6;
r.h.al=0;
r.h.ch=0;
r.h.cl=0;
r.h.dh=24;
r.h.dl=79;
r.h.bh=7;
int86(0x10,&r,&r);
}
```

যতবার স্ক্রিন ক্লিয়ার করার দরকার হবে ততবার ওপরের এই বড় প্রোগ্রামটি লিখতে হবে। কিন্তু লাইব্রেরি ফাংশন ব্যবহার করলে খুব সহজেই কাজটি করা সম্ভব। এরকম আরও বিভিন্ন কাজের জন্য বিভিন্ন লাইব্রেরি ফাংশন ডিফাইন করা আছে।

আরেক ধরনের ফাংশনের নাম হলো ইউজার ডিফাইন্ড ফাংশন। এর মাঝে আসলে তেমন কোনো পার্থক্য নেই। লাইব্রেরি ফাংশন হলো যেগুলো আগে থেকে বর্ণিত থাকে সেগুলো। আর ইউজার ডিফাইন্ড ফাংশন হলো যেগুলো ইউজার যে ফাংশনগুলো নিজের সুবিধার জন্য বানিয়ে নেয় সেগুলো। ইউজার ডিফাইন্ড ফাংশনের একটি উদাহরণ নিচে দেয়া হলো :

```
#include<stdio.h>
void func();
void main()
{
      clrscr();
func();
}
void func()
{
      printf("a user defined function is
created");
}
```

এখানে একটি ইউজার ডিফাইন্ড ফাংশন তৈরি করা হয়েছে, যার নাম func()। মেইন ফাংশনের মাঝে যখন এই ফাংশনকে কল করা হবে, তখন প্রোগ্রাম এই ফাংশনে চলে যাবে। সেখানে একটি লাইন প্রিন্ট করে প্রোগ্রাম আবার মেইন ফাংশনে ফিরে আসবে। তারপর আর কোনো কমান্ড নেই বলে প্রোগ্রাম টারমিনেট করবে। এভাবে মেইন ফাংশন থেকে ইউজার ডিফাইন্ড ফাংশনে কল করতে হয়।

ফাংশনের অনেক ধরনের ব্যবহার আছে। একটি ফাংশনকে বিভিন্নভাবে ব্যবহারের মাধ্যমে প্রোগ্রামকে অনেক সহজ করে তোলা যায়। ফাংশনের বিভিন্ন ব্যবহারের জন্য ফাংশনের বিভিন্ন অংশ সম্পর্কে জানা প্রয়োজন। একটি ফাংশনের প্রধানত তিনটি অংশ থাকে। যেমন : ফাংশন ডেফিনিশন, ফাংশন বডি, ফাংশন কলিং ইত্যাদি।

ফাংশন ডেফিনিশনকে তিন ভাগে ভাগ করা যায়। যেমন : ফাংশন নাম, প্যারামিটার বা আরগুমেন্ট লিস্ট ও রিটার্ন ডাটা টাইপ। ফাংশনের নাম হলো ইউজারের পছন্দ অনুযায়ী একটি নাম। তবে সে নাম নির্ধারণ করার কিছু নিয়ম আছে। ভেরিয়েবলের নাম লেখার নিয়মানুসারে ফাংশনের নাম নির্ধারণ করতে হয়। প্যারামিটার হলো একটি ফাংশনের গুরুত্বপূর্ণ অংশগুলোর একটি। ফাংশনের নামের পর যে () থাকে, সেখানে প্যারামিটার লিখতে হয়। প্যারামিটার সাধারণত কোনো ডাটা পাসিংয়ের জন্য ব্যবহার হয়। যেমন : মেইন ফাংশনের একটি ভেরিয়েবল i ডিক্লেয়ার করা হলো, যার মান ৫। একটি ইউজার ডিফাইন্ড ফাংশন আছে যার নাম func()। এখন ইউজার চাচ্ছেন func()-এ i-এর মান পাঠাতে। তাহলে সেক্ষেত্রে ফাংশনের প্যারামিটার ব্যবহার করতে হবে। প্যারামিটার দুই ধরনের হয়। যেমন : রিয়েল প্যারামিটার এবং ফরমাল প্যারামিটার। রিয়েল প্যারামিটার হলো যেই মানটি পাঠানো হচ্ছে। আর ফরমাল প্যারামিটার হলো কোনো ফাংশন যে ধরনের ডাটা গ্রহণ করতে পারে। পরে একটি উদাহরণে সেটি দেখানো হয়েছে। আর একাধিক প্যারামিটার ব্যবহার করতে চাইলে সেগুলো কমা (,) দিয়ে লিখতে হয়। এক্ষেত্রে খেয়াল রাখতে হবে ফরমাল প্যারামিটার যে সিরিয়ালে আছে রিয়েল প্যারামিটারও সেই সিরিয়ালে কাজ করবে।

রিটার্ন টাইপ হলো আসলে যেকোনো ডাটা টাইপ। এর মাধ্যমে ফাংশনকে বলে দেয়া হয় সে কোন ধরনের ডাটা রিটার্ন করবে। কোনো ফাংশনকে যখন কল করা হয়, তখন ফাংশনটি সেখানে একটি ভ্যালু সহকারে রিটার্ন করতে সক্ষম। যেমন : i=func(); লিখলে এর মানে হবে, এই ফাংশনটি কল করলে সেটি যে ভ্যালু নিয়ে রিটার্ন করবে তা i-এর মান হিসেবে নির্ধারিত হবে। কোনো ফাংশনের যদি কোনো ভ্যালু রিটার্ন করার প্রয়োজন না থাকে তাহলে রিটার্ন টাইপ void হয়। এছাড়া প্রয়োজনানুসারে int, double ইত্যাদি হতে পারে। এখানে এটি বিষয় খেয়াল রাখতে হবে, কোনো ফাংশনের রিটার্ন টাইপ থাকলেই সেটি ডাটা রিটার্ন করে না। বরং ফাংশনের ভেতরে রিটার্ন স্টেটমেন্ট থাকতে হবে। কিন্তু রিটার্ন স্টেটমেন্ট এবং রিটার্ন টাইপ একই হতে হবে। ভিন্ন হলে এরর দেখাবে। রিটার্ন টাইপ হলো ফাংশনটি কী ধরনের ডাটা নিয়ে রিটার্ন করতে সক্ষম সেটা বলে দেয়া। আর রিটার্ন স্টেটমেন্ট হলো এমন একটি স্টেটমেন্ট যার মাধ্যমে ফাংশনকে কোনো একটি ভ্যালু নিয়ে করার কমান্ড দেয়া হয়। আর কোনো ফাংশনের

(বাকি অংশ ৮৫ পৃষ্ঠায়)

প্রোগ্রামিং সি/সি++

(৮৩ পৃষ্ঠার পর)

যদি রিটার্ন স্টেটমেন্ট না থাকে তাহলে দ্বিতীয় বন্ধনী পাওয়ার সাথে সাথে ফাংশনটি রিটার্ন করবে।

ফাংশন ডেফিনিশন বা বডি হলো কোনো ফাংশনের মাঝে যা লেখা থাকে সেসব। অর্থাৎ একটি ফাংশন যেসব কাজ করবে তাই হলো ফাংশনের বডি। কোনো লাইব্রেরি ফাংশনের বডি নিয়ে ইউজারকে চিন্তা করতে হয় না। কারণ, তা বিল্টইন দেয়া থাকে। তবে নিজে কোনো ফাংশন তৈরি করলে সেই ফাংশনের বডি সঠিকভাবে লিখতে হয়, তা না হলে প্রোগ্রামে ভিন্ন আউটপুট আসতে পারে।

ফাংশনকে বিভিন্নভাবে ব্যবহার করা যায়। যেমন : একটি ইউজার ডিফাইন্ড ফাংশনকে শুধু কিছু সাধারণ কাজ করার জন্য ব্যবহার করা যেতে পারে, যে কাজগুলোর ওপর কোনো কিছু নির্ভরশীল নয়। আবার ফাংশনকে কোনো কিছুর মান নির্ধারণ করার কাজেও ব্যবহার করা যেতে পারে। যেমন : একটি ফাংশন তৈরি করা হলো যার ভেতরে কিছু হিসাব-নিকাশ করে একটা ভ্যালু রিটার্ন করা হলো। এখন ইউজার চান সেই রিটার্ন করা ভ্যালুটি i ভেরিয়েবলের মান হিসেবে নির্ধারণ হোক। তাহলে ফাংশনটিকে `i=func();` এভাবে কল করতে হবে। ফাংশনের বিভিন্ন প্রোগ্রাম করলে এ বিষয়টি আরও স্পষ্ট হবে।

ফিডব্যাক : *wahid_cseaust@yahoo.com*