সি প্রোগ্রামিং এ ফাংশন একটি গুরুত্বপূর্ণ শব্দ। বড় কোন প্রোগ্রাম করতে গেলেই আমরা ফাংশন ব্যাবহার করে থাকি। কারণ প্রত্যেকটা ফাংশন তার নিজের কমান্ড অনুযায়ী কাজ করে এবং সেই অনুযায়ী ফলাফল রিটার্ন করে থাকে।

সি তে ফাংশন হয় ২ ধরনের

১। লাইব্রেরি ফাংশন

২। ইউজার দিফাইন্দ ফাংশন

লাইব্রেরি ফাংশন হল সে গুলো যে গুলো আমরা #include < .h > দিয়ে লিখে থাকি। যেমনঃ বলা যায়

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include<conio.h>

#include<string.h>

#include<time.h>

এই লাইব্রেরি ফাংশনে আগে থেকে কমান্ড দেওয়া থাকে সেই অনুযায়ী কাজ হয়।

কিন্তু যে গুলো ইউজার দিফাইন্দ ফাংশন সে গুলো মুলত এক জন কোডার নিজের প্রয়োজন অনুযায়ী

লিখে থাকে বা তৈরি করে।

একটা ফাংশন যখন আমরা লিখবো , তখন যেই জিনিষ গুলো ফাংশনে থাকে সে গুলো হল

1. Function Name

2. Function Type

3. List Of Parameters

4. Variable Declarations

5. Return Statements

ফাংশন এর একটা নাম থাকবে। এটা ইছহা মত দেওয়া যায়। কিন্তু একটা ফাংশন যেই কাজ করবে ,সেই অনুযায়ী নাম দিলে ফাংশন এর কাজ খুব সহজে বুঝা সম্ভহব।

ফাংশন এর টাইপঃ এটা বলতে বুঝায় ফাংশনটা কি রকম ধরনের হবে? মানেঃ integer, float, double এদের ভিতর কোনটা ।

তাছাড়া

ফাংশন যদি কোন ভেলু রিটার্ন করে তাহলে তার আগে int লিখতে হয় । আর যদি কোন কিছু রিটার্ন না করে তাহলে void লিখা হয়।

int add(int n1, int n2)

এটা একটা ফাংশন এর লিখার ধরন । যেটা দ্বারা বুঝায় এটা রিটার্ন টাইপের ফাংশন। ফাংশন এর নাম হল

Add , জার কাজ ২ তা সংখ্যা যোগ করা। এখন, ফাংশন গুলো মানে যে গুলো ইউজার দিফাইন্দ ফাংশন সে, গুলো মেইন ফাংশন থেকে কল হয়। এখন মেইন ফাংশন যদি কোডের নিচে লিখা হয় , তাহলে ফাংশন গুলোর নাম ডিকলেয়ার করতে হয় না। কিন্তু যদি মেইন ফাংশন শুরু তে লিখা হয় তখন , মেইন ফাংশন এর উপরে ফাংশন গুলোর নাম এবং প্যারামিটার ডিকলেয়ার করতে হয় ।

int add(int n1, int n2)

ফাংশন এর ভিতর int n1. Int n2 হচ্ছে প্যারামিটার ।

একটা সহজ ফাংশন এর কোড আমরা এখন দেখবো।

#include<stdio.h>

1. int addition(int first, int second);

2. main()

3. {

4. int first, second, sum;

5. scanf("%d%d", &first, &second);

6. sum = addition(first, second);

7. printf("%d**\n**", sum);

8. return 0;

}

9. int addition(int a, int b)

{

10. int result;

11. result = a + b;

12. return result;

}

১ নাম্বার লাইনে অ্যাড ফাংশনের নাম ও প্যারামিটার

লিখা হয়েছে, কারণ মেইন ফাংশন শুরুতেই আছে ২ নাম্বার লাইনে। মাইন ফাংশন এর ভিতর int first, second, sum; এই গুলো হল লোকাল ভেরিএবেল। যারা শুধু মেইন ফাংশন এর জন্য কাজ করবে। মেইন ফাংশন থেকে ইনপুট নেওয়া হবে ২ টা সংখ্যা। যেটা করা হয়েছে ৫ নাম্বার লাইনে। এরপর ২ টা সংখ্যা যোগ করার ভেরিয়েবেল sum কে ৬ নাম্বার লাইনে লিখা হয়েছে। এখন এই ৬ নং লাইনে sum এর পর সমান সাইন দিয়ে addition লিখা হয়েছে , যেটা হল ফাংশন কলিং। ধরলাম আমি একটা অফিসে কাজ করি মেইন ফাংশন হল আমার বস। আমি হলাম . sum ভেরিএবেল তো আমার বস আমাকে ডেকে বল্ল ২ টা সংখ্যা যোগ করো। এখন আমি নিজে যোগ করবো না , আমি খুব অলস, তো আমি যেটা করতে চাই সেটা হল অফিসে addition নামে আতেল কেউ একজন আছে যে এই কাজ টা করতে পারে। আমি এখন addition কে ডেকে ২ টা সংখ্যা দিয়ে বলবো যোগ করো। addition এর নিজের রুম আছে, সে ধরলাম নাম্বার ২ টা নিয়ে তার রুমে গেল যোগ করতে। যেটা ৯ নাম্বার লাইনে । আমি ৬ নং লাইনে তাকে ২ টা সংখ্যা দিয়েছি। সে ধরলাম সংখ্যা ২ টি পকেটে করে নিয়ে গেলো। এখন আমরা যখন কোন কিছু পকেটে রাখি তার আগে জেনে নেই জিনিষ টা কি ধরনের, সংখ্যা ২ টা কি ধরনের সেটাই 9. int addition(int a, int b) int a , int b দিয়ে বুঝানো হল।

এখন

Addition যেটা করবে সেটা হল সংখ্যা ২ টা যোগ করবে কিন্তু কিভাবে? ধরলাম সে সংখ্যা ২ টা যোগ করে একটা ভেরিএবেলে রাখবে যেটার নাম int result;

যোগ শেষ , এই বার Addition এর কাজ হল ফলাফল আমার কাছে দেওয়া, আর আমার কাজ বস কে ফলাফল জানিয়ে দেওয়া। এখন addition যা করবে সেটা হল . result আমার কাছে দিবে কিন্তু সে নিজে এসে দিবে না। 12. return result; এই লাইন এর মানে হল একটা ভেলু রিটার্ন করা। এখন কথা হল কাকে , কিভাবে রিটার্ন করবে। result; হল addition ফাংশন এর ভেরিয়েবেল। তাই সে

Addition ফাংশন কে result রিটার্ন করবে। যেটা হল একটা integer । এখন addition কে কল করেছিলাম আমি মেইন ফাংশন থেকে , তাই যেটা হবে ,

6. sum = addition(first, second);

এইখানে addition কে

Result যে integer ভেলু রিটার্ন করে সেইটা চলে আসবে। মানে মিস্টার addition

আমাকে যোগ ফল দিয়ে দিয়েছে। এখন আমার কাজ হল ফলাফল বস কে জানানো। আমি ফলাফল পাচ্ছি . sum । যেটা 7. printf("%d**\n**", sum);

তে প্রিন্ট করে দেওয়া হল বস অর্থাৎ মেইন ফাংশন এর নির্দেশে।

ইউজার দিফাইন্দ ফাংশন গুলো ৪ ধরনের হয়।

1. [Function with no arguments and no return value](http://www.programiz.com/c-programming/types-user-defined-functions#no_no)
2. [Function with no arguments and return value](http://www.programiz.com/c-programming/types-user-defined-functions#no_yes)
3. [Function with arguments but no return value](http://www.programiz.com/c-programming/types-user-defined-functions#yes_no)
4. [Function with arguments and return value](http://www.programiz.com/c-programming/types-user-defined-functions#yes_yes).

1. [Function with no arguments and no return value](http://www.programiz.com/c-programming/types-user-defined-functions#no_no) :

আরগুমেন্ত মানে হল ফাংশন এর প্যারামিটার, এই টাইপ এর ফাংশনে কোন প্যারামিটার থাকে না। আর এই টাইপ এর ফাংশন কোন কিছু রিটার্ন করে না।

একটা কোড দেখলেই আমরা বুঝতে পারবো ।

#include <stdio.h>

void prime();

int main(){

prime(); //No argument is passed to prime().

return 0;

}

void prime(){

/\* There is no return value to calling function main(). Hence, return type of prime() is void \*/

int num,i,flag=0;

printf("Enter positive integer enter to check:\n");

scanf("%d",&num);

for(i=2;i<=num/2;++i){

if(num%i==0){

flag=1;

}

}

if (flag==1)

printf("%d is not prime",num);

else

printf("%d is prime",num);

}

void prime এই ফাংশন কোন ভেলু রিটার্ন করে না। void prime() ফার্স্ট ব্র্যাকেট এর ভিতর কোন প্যারামিটার

নেই ।

1. [Function with no arguments and return value](http://www.programiz.com/c-programming/types-user-defined-functions" \l "no_yes)-

এই টাইপের ফাংশনে কোন প্যারা মিটার থাকে না , তবে এটা রিটার্ন করতে পারে।

একটা কোড দেখলেই আমরা বুঝতে পারবো-

int input();

int main(){

int num,i,flag = 0;

num=input(); /\* No argument is passed to input() \*/

for(i=2; i<=num/2; ++i){

if(num%i==0){

flag = 1;

break;

}

}

if(flag == 1)

printf("%d is not prime",num);

else

printf("%d is prime", num);

return 0;

}

int input(){ /\* Integer value is returned from input() to calling function \*/

int n;

printf("Enter positive integer to check:\n");

scanf("%d",&n);

return n;

}

int input() ফার্স্ট ব্র্যাকেট এর ভিতর কোন প্যারা মিটার নেই। কিন্তু int input () এই ফাংশন এর শেষে return n;

দেওয়া আছে । যেটা মেইন ফাংশন এর num=input(); কে একটা ইনটিজার রিটার্ন করে থাকে।

3.[Function with arguments but no return value](http://www.programiz.com/c-programming/types-user-defined-functions" \l "yes_no)-

এই টাইপের ফাংশনে প্যারামিটার মানে আর্গুমেন্ট দেওয়া থাকে। তবে এটায়

রিটার্ন লিখা হয় না

একটা কোড দেখলেই আমরা বুঝতে পারবো।

#include <stdio.h>

void check\_display(int n);

int main(){

int num;

printf("Enter positive enter to check:\n");

scanf("%d",&num);

check\_display(num); /\* Argument num is passed to function. \*/

return 0;

}

void check\_display(int n){

/\* There is no return value to calling function. Hence, return type of function is void. \*/

int i, flag = 0;

for(i=2; i<=n/2; ++i){

if(n%i==0){

flag = 1;

break;

}

}

if(flag == 1)

printf("%d is not prime",n);

else

printf("%d is prime", n);

}

void check\_display(int n) এই ফাংশনের ফার্স্ট ব্র্যাকেট এর ভিতর প্যারামিটার আছে।

void check\_display এই ফাংশনের শেষে কোন return জাতীয় কিছু লিখা নেই। তার মানে এই টাইপ রিটার্ন করে না।

4 .[Function with arguments and return value](http://www.programiz.com/c-programming/types-user-defined-functions" \l "yes_yes).

এই টাইপ টাই সব থেকে বেশি ব্যাবহার করা হয়। এদের প্যারামিটার থাকে , সাথে থাকে return ।

একটা কোড দেখলেই আমরা বুঝতে পারবো

#include <stdio.h>

int check(int n);

int main(){

int num,num\_check=0;

printf("Enter positive enter to check:\n");

scanf("%d",&num);

num\_check=check(num); /\* Argument num is passed to check() function. \*/

if(num\_check==1)

printf("%d is not prime",num);

else

printf("%d is prime",num);

return 0;

}

int check(int n){

/\* Integer value is returned from function check() \*/

int i;

for(i=2;i<=n/2;++i){

if(n%i==0)

return 1;

}

return 0;

}

int check(int n) ফাংশনে প্যারা মিটার আছে ফার্স্ট ব্র্যাকেট এর ভিতর। int check এই ফাংশন return করে।

কেন আমরা return 0 ; লিখি?

আমাদের ফাংশন যদি int দিয়ে শুরু হয় মানে, int main()

তখন আমরা return 0 ; লিখে থাকি।

যদি ফাংশন হয় void দিয়ে , মানে void main()

তাহলে আমরা return 0 ;

লিখি না ।

return; , return a; , return 1; , return -1; এগুলোর কাজ কি? (a হল কোন একটা ভেরিএবেলের নাম)

শুধু return; এটা কোন ভেলু রিটার্ন

করে না।

এটা একটা ফাংশন এর closing brace হিসেবে কাজ করে থাকে।

যদি কোন কোড এ শুধু return; লিখা থাকে, তাহলে যেটা হয় যখন return; এর লাইন আসে , তখন কোড এর কন্ট্রোল কলিং ফাংশন এর কাছে চলে যায়।

return a;

এটা নির্দিষ্ট ফাংশন কে অপারেশন শেষে ভেলু রিটার্ন করে দেয়।

একটা কোড দেখলেই আমরা বুঝতে পারবো।

#include<stdio.h>

int add(int a,int b);

int main()

{

int x,y,sum;

printf("Enter First Number\n");

scanf("%d",&x);

printf("Enter Second Number\n");

scanf("%d",&y);

sum=add(x,y);

printf("The Sum Is :%d\n",sum);

return 0;

}

int add(int c,int b)

{

int a;

a=c+b;

return a;

}

int add(int c,int b)

{

int a;

a=c+b;

return a;

}

এখানে return a; int add ফাংশন কে একটা ইনটিজার রি টার্ন

করে থাকে। মেইন ফাংশনে int add

ফাংশন কল হয়েছে, সে জন্যে ফাইনালি মেইন ফাংশন a এর

ভেলু ক্যাচ করে।

একটা ফাংশনে একের অধিক return

স্টেটমেন্ট থাকতে পারে। ।

ফাংশন বাই ডিফল্ট ইনটিজার টাইপের ডাটা রিটার্ন করে থাকে।

আবার যখন কোন ফাংশন কোন ভেলু রিটার্ন করে তখন সেটা ফাংশন এর টাইপ অনুযায়ী কাস্টিং হয়ে যায়।

যেমনঃ

int multiplication ( )

{

return (2.5 \* 3.0);

}

এই ফাংশন টায় ২ টা float ভেলু গুন করা হয়েছে।

কিন্তু এদের রিটার্ন

করার কথা ৭.৫ , কিন্তু কাস্টিং এর কারণে শুধু ৭ রিটার্ন করবে।