

C Tutorial



What is Programming?

- Introduction
- Installation
- Tokens & Identifiers
- Data Types
- Variables
- First Program
- Flow Chart
- Operators
- Decision Making
- Looping
- Arrays
- Strings
- Functions
- Structures
- Unions
- Recursion
- Pointers
- Type Casting
- Storage Classes
- Console I/O
- File Handling
- Error Handling
- Header Files
- Preprocessors

What is Programming?

- Introduction to **programming in Hindi**
- *Programming language in Hindi*
- What is Compiler in Hindi
- Who is Programmer in Hindi

Introduction to Programming

India में हर साल लाखों students computer scientists बनते हैं। हालाँकि एक computer scientist सिर्फ programming ही नहीं करता है, लेकिन programming computer science का बहुत ही महत्वपूर्ण part है। इसलिए एक शुरुआती programmer होने के नाते आपके लिए programming और उससे related components का concept ठीक से समझना बहुत जरूरी है। कितने ही ऐसे programmers होते हैं जो programming तो कर रहे होते हैं लेकिन उनको programming के पीछे का concept नहीं पता होता है। आइये इसके बारे में जानने का प्रयास करते हैं।

आसान शब्दों में कहा जाये तो programming एक ऐसी process होती है जिसमें एक programmer CPU (Central Processing Unit) को कोई task perform करने के instructions देता है। ये task कुछ भी हो सकता है जैसे की दो numbers को add करना आदि।

Programming कुछ components से मिलकर एक समग्र रूप धारण करती है।

1. Programming Language
2. Compiler / Interpreter
3. Programmer

इन components के बारे में निचे detail से बताया जा रहा है।

Programming Language

Humans जब भी interact करते हैं तो किसी language के माध्यम से करते हैं। एक CPU को भी आप language के माध्यम से ही instructions देते हैं। ऐसी language जो CPU को instructions देने के लिए programmers यूज़ करते हैं programming language कहलाती है।

CPU सिर्फ binary language (0 & 1) को ही समझता है। लेकिन किसी भी human के लिए binary language में programming करना असंभव है। इसलिए simple English sentences को यूज़ करते हुए programming languages develop की गयीं।

ऐसा करने से programming करना बहुत अधिक आसान हो गया। अब जैसा की आप देख सकते हैं हर कोई basic information प्राप्त करके programming करना सिख सकता है। लेकिन पहले ऐसा संभव नहीं था।

Types of Programming Languages

अब तक 256 से अधिक programming languages develop की जा चुकी हैं। हर language को एक different purpose के लिए develop किया गया है। Purpose के according programming languages को mainly 5 categories में divide किया गया है।

1. Machine Language / Machine Code - ये वो code होता है जिसे CPU directly execute करता है। Machine code hexadecimal में होता है। ये hardware के बहुत ही करीब होता है। इसे low level language भी कहा जाता है।



2. Assembly Language - Machine code को और भी अधिक readable बनाने के लिए assembly language create की गयी। इसमें labels को add किया गया है जिससे tasks perform करना और भी आसान हो जाता है। ये भी एक low level language होती है।
3. High-Level Language - High level languages hardware से independent होती है जैसे की c, c++, java आदि। इन languages का code directly machine code में convert नहीं होता है।
4. System Language - इस प्रकार की languages computer को manage करने के लिए यूज़ की जाती है। इन्हें low level tasks perform करने के लिए यूज़ किया जाता है। जैसे की memory management, process management आदि।
5. Scripting Languages - इस प्रकार की languages system और दूसरी applications के बीच में bridge की तरह काम करती है। जैसे की PHP आदि।

Some Common Programming Languages

यहाँ पर कुछ programming languages की list दी जा रही है। इनमें से कुछ के बारे में आप सुन चुके होंगे।

1. C Language
2. C++
3. Java
4. C#
5. PHP
6. HTML
7. JavaScript
8. Pascal
9. Visual Basic
10. FORTRAN

Compiler / Interpreter

जैसा की मैंने आपको बताया computer सिर्फ binary language समझता है और humans का binary में programming करना बहुत मुश्किल है। इसलिए एक ऐसा program develop किया गया जो programmers से simple English statements के रूप में instructions ले सके और उसे binary में convert करके CPU को pass कर दे। इस program को compiler कहते हैं।

Compiler एक programmer और CPU के बीच bridge की तरह काम करता है। इसे आप translator भी कह सकते हैं। एक compiler बहुत से tasks perform करता है इनके बारे में निचे दिया जा रहा है।

- Compiler syntax error check करता है। यदि कोई syntax error होती है तो इसके बारे में programmer को notify किया जाता है।
- Compiler source code को machine language में convert करता है।
- Compiler एक machine executable file generate करता है।

आइये अब interpreter के बारे में जानने का प्रयास करते हैं।

कुछ languages compiler की जगह interpreter भी use करती है। Compiler और interpreter दोनों एक ही काम करते हैं। फर्क सिर्फ इतना होता है की compiler पुरे source code को एक साथ compile करके machine code generate करता है जिसे एक बार में completely execute किया जा सकता है। जबकि एक interpreter एक एक instructions को machine code में convert करता है और साथ की साथ execute करता जाता है। अपनी need के according languages compiler और interpreter को यूज़ करती है।

Programmer

शुरुआती दिनों में लोग interest की वजह से programming करते थे लेकिन अब इसने एक job का रूप धारण कर लिया है। Programmer वह person होता है जिसे की किसी programming language का ज्ञान होता है। एक programmer अपने ज्ञान को यूज़ करते हुए programming language की मदद से real world problems को solve करता है।

Best Hindi Tutorials

Computer Science and IT tutorials in Hindi

[Home](#) [About](#) [Contact](#) [Recommend](#)



Build scalable
Machine Learning models

[TRY IT FREE](#)

C Language

What is Programming?

C - Introduction

C - Tokens, Identifiers & Keywords

C - Data Types

C - Variables

C - First Program

C - Flow Chart

C - Operators

C - Decision Making

C - Looping

C - Arrays

C - Strings

C - Functions

C - Structures

C - Unions

C - Recursion

C - Pointers

C - Type Casting

C - Storage Classes

C - Console I/O

C - File Handling

C - Error Handling

C - Header Files

C - Preprocessor Directives

C in Hindi

- Introduction to **C in Hindi**
- Characteristics & Features of *C in Hindi*
- Limitations of C in Hindi
- Applications of C in Hindi
- A simple C program

Introduction to C language

C एक बहुत ही popular programming language है। इसे Middle level language माना जाता है। आज भी C की popularity में कोई कमी नहीं आई है। किसी भी दूसरी language को सिखने से पहले C को सीखना बहुत जरूरी है। क्योंकि C के बाद जितनी भी **programming languages** आई उन्होंने किसी न किसी तरह C के features को adopt किया है। आइये C के बारे में detail से जानने का प्रयास करते हैं।

History

C एक **structured** general purpose programming language है। इसे **UNIX operating system** को दुबारा लिखने के लिए develop किया गया था। इससे पहले UNIX operating system को B language में लिखा गया था। पूरा UNIX operating system और उस पर run होने वाले ज्यादातर programs C language में ही लिखे गए हैं।

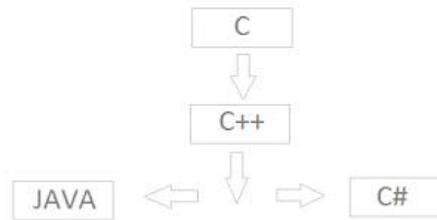
B language Ken Thompson के द्वारा 1970 में Bell laboratories में लिखी गई थी। B एक type less language थी। C language को **Dennis Ritchie** ने 1972 में **Bell laboratories** में ही develop किया था। C language बहुत से data types provide करती है।

C language के syntax को बहुत सी popular languages ने follow किया है। C का improved version C# है। C के features को Java और C++ जैसी languages ने follow किया है।

Google द्वारा विज्ञापन बंद कर दिया गया

यह विज्ञापन देखना बंद करें

Google द्वारा विज्ञापन ①



Characteristics & Features

- C एक **modular** language है। C में एक बड़े program को छोटे छोटे modules (functions) के रूप में divide किया जाता है। ऐसा करने से programs को manage और debug करने में आसानी रहती है।
- C एक flexible language है। C 32 reserved keywords provide करती है। ये keywords programmer को language पर control provide करते हैं। इन keywords की मदद से program के structure को भी modify किया जा सकता है।
- C language Assembly language के बाद सबसे fast language मानी जाती है। इसे कई बार low level language भी कहा जाता है। C language hardware के बहुत करीब होती हैं। इसलिए ये दूसरी programming languages से fast होती हैं।
- C एक case sensitive language है।
- C High level और low level दोनों तरह के programs create करने में सक्षम है।
- C एक extendable language है। C programs में libraries के द्वारा आप कभी भी नए features add कर सकते हैं।

Limitations

- C language में run time type checking नहीं होती है। C language run time में किसी भी variable के data type को identify करने में असमर्थ होती है। उदाहरण के लिए float data type में integer value डाली जा सकती है।
- C language में object oriented concepts नहीं होते हैं।
- C language में constructors और destructors का concept भी नहीं होता है।
- C language में namespaces का concept नहीं होता है। इसलिए different programmers के code को एक ही project में यूज करना बहुत मुश्किल होता है।
- C language code की re-usability को support नहीं करती है।
- C language में exceptions को run time में handle नहीं किया जा सकता है।

Applications

- ज्यादातर operating systems C language में ही बनाए जाते हैं। जैसे की UNIX, Windows आदि।
- किसी भी नयी programming language को compile करने के लिए compilers की आवश्यकता होती है। C language में **compilers** भी create किये जाते हैं। उदाहरण के लिए java का compiler javac भी C language में ही बनाया गया है।
- C language में कई प्रकार के assemblers भी create किये जाते हैं। Assembler basic instructions को bits में convert करते हैं।
- C language में text editors भी create किये जा सकते हैं। जैसे की gedit, आदि।
- C language में network drivers भी create किये जाते हैं।
- C language में database systems भी लिखे जाते हैं। जैसे की oracle database C में ही लिखा गया है।

Best Hindi Tutorials

Computer Science and IT tutorials in Hindi

[Home](#) [About](#) [Contact](#) [Recommend](#)



Put Ads on your blog with AdSense

[LEARN MORE](#)

C Language

What is Programming?

C - Introduction

C - Tokens, Identifiers & Keywords

C - Data Types

C - Variables

C - First Program

C - Flow Chart

C - Operators

C - Decision Making

C - Looping

C - Arrays

C - Strings

C - Functions

C - Structures

C - Unions

C - Recursion

C - Pointers

C - Type Casting

C - Storage Classes

C - Console I/O

C - File Handling

C - Error Handling

C - Header Files

C - Preprocessor Directives

C Data Types

- Introduction to **C data types in Hindi**
- Different *C data types in Hindi*
 - Integer type in Hindi
 - Floating point type in Hindi
 - Char type in Hindi

Introduction To C Data Types

जब भी आप कोई variable create करते हैं तो उससे पहले compiler को बताते हैं की आप किस तरह का data उस variable में store करेंगे। इससे compiler उतनी ही memory उस variable को computer की memory में से allot कर देता है। जैसे की निचे दिए गए statement को देखिये।

```
int age;
```

ऊपर दिया गया statement compiler को बताता है की आप age variable में एक whole number store करने वाले हैं। किसी भी whole number को store करने के लिए जितनी memory की आवश्यकता होती है compiler उतनी इस variable को allot कर देता है। जो की 2 bytes है।

अब आप सोच सकते हैं की यदि किसी programming language में data types ना हो तो कितनी memory waste हो सकती है। जहाँ पर 2 bytes की requirement हो वहाँ 20 bytes आप waste कर सकते हैं।

इस प्रकार जितनी भी तरह का data आप store कर सकते हैं उसके लिए पहले से ही maximum memory limit define की गई है। आइये अब C के अलग अलग तरह के data types के बारे में जानने का प्रयास करते हैं।

Different C data types

C language 3 तरह के data types को support करती है। इन्हें primitive types भी कहते हैं।

- Integer



- int
- short int
- long int
- signed int
- unsigned int
- Floating point
 - float
 - double
- Character
 - char

Integer

Integer types किसी भी whole number (बिना दशमलव के) को store करने के लिए यूज़ किये जाते हैं। Integer types 5 प्रकार के होते हैं। हालांकि ये सभी whole number को store करते हैं। लेकिन memory size और range के base पर इन्हें categories किया गया है।

Data type	Size (Bytes)	Range
int	2	-32768 से 32767
short int	1	-128 से 127
long int	4	-2,147,483,648 से 2,147,483,647
signed int	2	-32768 - 32767
unsigned int	2	0 - 65535

Example

```
long int population = 200000000;
```

Floating point

Floating point data types को दशमलव संख्याओं को store करने के लिए define किया गया है। Floating point data types 2 तरह के होते हैं। इनको size और range के base पर categories किया गया है। Float type में आप दशमलव के बाद 7 digits तक store कर सकते हैं। Double type में दशमलव के बाद 17 digits तक store की जा सकती है।

Data types	Size (Bytes)	Range
Float	4	3.4E-38 से 3.4E+38
Double	8	1.7E-308 से 1.7E+308

Example

```
double balance=810.12354984;
```



Character

Character type को एक character store करने के लिए यूज़ किया जाता है। इनको 2 categories में divide किया गया है।

Data type	Size (Bytes)	Range
char	1	-128 से 127
unsigned char	1	0 से 255

Example

```
char bestlanguage = "C";
```

Void type

Void type को उन situations में यूज़ किया जाता है जब आपको value के बारे में कोई जानकारी ना हो। इसे functions के साथ ज्यादातर यूज़ किया जाता है। C में void type को आप इन situations में यूज़ कर सकते हैं।

1. यदि आपका function कोई value return नहीं करता है तो आप उसका return type void define करते हैं। उदाहरण के लिए आप इस प्रकार function define कर सकते हैं।
void myFunction();
2. यदि आप function में कोई parameters नहीं ले रहे हैं तो आप वहाँ पर void define कर सकते हैं। Void type से पता चलता है की इस function में कोई argument नहीं लिया जाता है। उदाहरण के लिए आप इस प्रकार void को parameter के रूप में pass कर सकते हैं।
int myFunction(void);
3. यदि आप sure नहीं हैं की pointer variable किस type के variable को point करेगा तो आप उसका type void declare कर सकते हैं। इसके बाद आप void pointer से किसी भी variable को point कर सकते हैं।

© Best Hindi Tutorials 7/18/2016



4 Replies so far - Add your comment

Raj Kumar

सोमवार, 02 जनवरी, 2017

Integer and character in dono ki limit mujhe samjh me nhi aaya
Like jaise aap ne Floating point me limit 7 and 17 hai wese in dono ke bare me explain kre plz

उत्तर दें

उत्तर

Vipin Sharma

सोमवार, 02 जनवरी, 2017



Character data type sirf ek character ko store karta h jese ki "a".
Or Integer types ki jitni range upar di gyi h usi range me aap koi
bhi sankhya us type ke variable me store kar sakte h.

उत्तर दें

Shubham Gupta

शनिवार, 18 फरवरी, 2017

Very nice article... thanks for sharing

उत्तर दें

उत्तर

Best Hindi Tutorials

शनिवार, 18 फरवरी, 2017

You are welcome! Thanks..

उत्तर दें

अपनी टिप्पणी लिखें...

इस रूप में टिप्पणी करें: Google खाता

प्रकाशित करें

पूर्वावलोकन



Best Hindi Tutorials

Computer Science and IT tutorials in Hindi

[Home](#) [About](#) [Contact](#) [Recommend](#)



Build scalable
Machine Learning models

[TRY IT FREE](#)

C Language

What is Programming?

C - Introduction

C - Tokens, Identifiers & Keywords

C - Data Types

C - Variables

C - First Program

C - Flow Chart

C - Operators

C - Decision Making

C - Looping

C - Arrays

C - Strings

C - Functions

C - Structures

C - Unions

C - Recursion

C - Pointers

C - Type Casting

C - Storage Classes

C - Console I/O

C - File Handling

C - Error Handling

C - Header Files

C - Preprocessor Directives

C operators

- Introduction to **C operators in Hindi**
- Types of *C operators in Hindi*
 - Arithmetic operators in Hindi
 - Relational operators in Hindi
 - Logical operators in Hindi
 - Bit wise operators in Hindi
 - Assignment operators in Hindi
 - Conditional operator in Hindi
 - Increment/Decrements operators in Hindi
 - Special C operators in Hindi

Introduction to C operators

Variables के अंदर values store करवाकर आप उन variables पर कई प्रकार के operations perform कर सकते हैं। उदाहरण के लिए दो integer variables के अंदर value store करवाकर आप addition का operation perform कर सकते हैं और उन दोनों variables की values के sum को print करवा सकते हैं। इसी प्रकार आप और भी अलग अलग operations variables के साथ perform कर सकते हैं।

Variables के साथ operations perform करने के लिए आपको अलग अलग operators यूज़ करने पड़ते हैं। इस chapter में आपको ऐसे ही operators के बारे में बताया जा रहा है।

Operations में operators के साथ जो variables यूज़ होते हैं वो operand कहलाते हैं। उदाहरण के लिए नीचे दिए गए statement को देखिये।

```
c = a + b;
```

उपर दिए गए statement में a और b को operator (+) के साथ यूज़ किया गया है इसलिए ये दोनों variables operands कहलायेंगे।





Operators 2 प्रकार के होते हैं।

- Unary - इस प्रकार के operators सिर्फ एक ही operand के साथ यूज किये जाते हैं।
- Binary - इस प्रकार के operators के साथ 2 operands यूज किये जाते हैं।

आइये अब C language में यूज होने वाले अलग अलग operators के बारे में जानने का प्रयास करते हैं।

Arithmetic operators

Arithmetic operators mathematical operations perform करने के लिए यूज किये जाते हैं। जैसे की addition, subtraction, division और multiplication आदि। Arithmetic operators 5 प्रकार के होते हैं। ये basic mathematical operators होते हैं।

Operators	Description
+ (Addition)	ये operator दो variables की values को add करता है।
- (Subtraction)	ये operator एक variable की value में से दूसरे variable की value subtract करता है।
* (Multiplication)	ये operator 2 variables की values को multiply करता है।
/ (Division)	ये operator एक variable की value से दूसरे variable की value को divide करता है।
% (Modulus)	ये operator division के बाद शेष बची हुई value को प्राप्त करने के लिए यूज किया जाता है।

Relational operators

Relational operators दो variables की values को compare करने के लिए यूज किये जाते हैं। जैसे की आप इन operators का यूज करके पता लगा सकते हैं की क्या किन्हीं दो variables की values बराबर है और यदि बराबर नहीं है तो कौनसे variable की value बड़ी है और कौनसे variable की value छोटी है।

इस तरह के operators को conditional statements (if,if-else,switch,for,while आदि) के साथ यूज किया जाता है। ये operators condition check करने के लिए यूज किये जाते हैं। Condition true होने पर value true हो जाती है और condition false होने पर value false हो जाती है।

C language में यूज होने वाले सभी relational operators के बारे में नीचे दिया जा रहा है।

Operators	Descriptions
== (Equal To)	ये operator check करता है की क्या दोनों variables की values equal है।
!= (Not Equal To)	ये operator check करता है की क्या दोनों variables की values non equal है।
< (Lesser Than)	ये operator check करता है की क्या left operand की value



	right operand से छोटी है।
> (Greater Than)	ये operator check करता है की क्या left operand की value right operand से बड़ी है।
<= (Lesser than equal)	ये operator check करता है की क्या left operand की value right operand से छोटी या उसके समान है।
>= (Greater than equal)	ये operator check करता है की क्या left operand की value right operand से बड़ी या उसके समान है।

Logical operators

Logical operators को decision making statements के साथ यूज किया जाता है। ये operators control statements में एक साथ दो condition को check करने के लिए यूज किये जाते हैं। उदाहरण के लिए आप किसी if statement में एक की जगह 2 conditions check कर सकते हैं। Logical operators के बारे में नीचे दिया जा रहा है।

Operators	Description
&& (AND)	जब दोनों conditions true होती हैं तब control statement की value true हो जाती है।
(OR)	जब कोई भी एक condition true हो तो भी control statement की value true हो जाती है।
! (NOT)	ये operator एक ही condition के साथ यूज किया जाता है। जब वो condition false होती है तो control statement की value true हो जाती है।

Bit wise operators

Bit wise operators दिए गए variables पर bit level operations perform करने के लिए यूज किये जाते हैं। Variables की decimal values bits में convert की जाती है। इसके बाद उन bits पर operations perform किये जाते हैं।

C language में यूज होने वाले bit wise operators के बारे में नीचे दिया जा रहा है।

Operators	Description
& (Bit wise AND)	इस operator के द्वारा दोनों variables की same position वाली bits के साथ AND operation perform किया जाता है।
(Bit wise OR)	इस operator के द्वारा दोनों variables की same position वाली bits के साथ OR operation perform किया जाता है।
~ (Bit wise NOT)	ये operator सिर्फ एक ही operand के साथ यूज किया जाता है। जिस variable के साथ इसे यूज किया जाता है उस variable की



	value की सभी bits opposite हो जाती है। जैसे की 0 है तो 1 हो जाती है और 1 है तो zero हो जाती है।
^ (XOR)	ये एक special type का OR operator होता है। ये operator opposite bits होने पर 1 return करता है और same bits होने पर 0 return करता है।
<< (Left Shift)	ये operator left side के variable की bits को right side के variable में दी गयी value जितना left में shift करता है।
>> (Right Shift)	ये operator left side के variables की bits को right side के variable में दी गयी value जितना right में shift करता है।

जैसा की आपको पता है bit wise operators bits के साथ work करते है। मान लीजिये आपने program में 2 variables a और b create किये हुए है। इन दोनों variables में आपने क्रमशः 3 और 5 values store करवाई हुई है। इनकी bits पर work करने के लिए सबसे पहले आप इन्हें binary में convert कर सकते है।

3 = 0000011
5 = 0000101

सिर्फ आपको समझाने के purpose से इन्हें binary में convert किया गया है। आपको program में binary values insert करवाने की आवश्यकता नहीं है। Computer information को binary form में ही store करता है। ऊपर define किये गए सभी operations इन bits पर ही perform किये जायेंगे।

Assignment operators

Assignment operators variables की values को एक दूसरे को assign करने के लिए यूज किये जाते है। C language में यूज होने वाले विभिन्न assignment operators के बारे में नीचे दिया जा रहा है।

Operators	Description
=	ये operator right side के operand की value को left side के operand को assign करता है।
+=	ये operator left side के operand में right side के operand की value को add करके result left side वाले operand को assign करता है।



	इसे आप इस प्रकार भी लिख सकते हैं। $a = a+b;$
--	ये operator left side के operand की value में से right side के operand की value को subtract करके result left side के variable में store करवाता है। इसे आप इस प्रकार भी लिख सकते हैं। $a=a-b;$
*=	ये operator left side के operand की value को right side के operand की value से multiply करके result को left side के operand में store करता है।
/=	ये operator left operand की value को right operand की value से divide करके result को left side के operand में store करता है।
%=	ये operator left side के operand की value को right side के operand की value से divide करके शेष बचे हुए result को left side के operand में store करता है।

Increment/Decrements operators

किसी भी variable की value को एक number से instantly increase या decrease करने के लिए आप increment/ decrements operators यूज़ कर सकते हैं। इनके बारे में नीचे दिया जा रहा है।

Operators	Description
++ (increment)	ये एक unary operator होता है। ये operand की value को एक number से increase करता है। जब यह operand के पहले लगता है तो value पहले increment होती है और बाद में यूज़ की जाती है। जब ये operator operand के बाद लगता है तो operand की value यूज़ होने के बाद increment होती है।
--(decrements)	ये भी एक unary operator होता है। ये operand की value को एक number से decrease करता है। जब इसे operand से पहले यूज़ किया जाता है तो value यूज़ होने से पहले decrease होती है। जब इसे variable के बाद यूज़ किया जाता है तो value पहले यूज़ होती है और बाद में decrease होती है।

Conditional (?:) operator



Conditional operator को ternary operator भी कहा जाता है। ये if-else statement की short form होती है। इसका general structure इस प्रकार होता है।

```
condition ? stmt1 : stmt 2;
```

यदि condition true हो तो statement one return होगा नहीं तो statement 2 return होगा। इसका उदाहरण निचे दिया जा रहा है।

```
5>3 ? true : false;
```

Special C operators

Operators	Description
sizeof (var_name)	ये operator memory में variables की size return करता है।
&	ये operator variables का memory address return करता है।
*	ये operator variable का pointer return करता है।

© Best Hindi Tutorials

9/01/2016



Leave a comment

अपनी टिप्पणी लिखें...

इस रूप में टिप्पणी करें: Google खाता

प्रकाशित करें

पूर्वावलोकन



Best Hindi Tutorials

Computer Science and IT tutorials in Hindi

[Home](#) [About](#) [Contact](#) [Recommend](#)



C Language

What is Programming?

C - Introduction

C - Tokens, Identifiers & Keywords

C - Data Types

C - Variables

C - First Program

C - Flow Chart

C - Operators

C - Decision Making

C - Looping

C - Arrays

C - Strings

C - Functions

C - Structures

C - Unions

C - Recursion

C - Pointers

C - Type Casting

C - Storage Classes

C - Console I/O

C - File Handling

C - Error Handling

C - Header Files

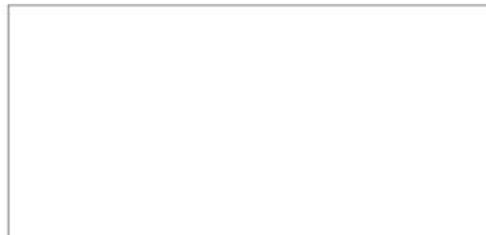
C - Preprocessor Directives

C variables & constants

- Introduction to **C variables in Hindi**
- Creating *C variables in Hindi*
- Scope of C variables in Hindi
- Introduction to C constants in Hindi

Introduction to C variables

यदि आप computer की memory में data को store करना चाहते हैं तो इसके लिए पहले आपको उस memory location के लिए नाम देना होता है। Variable memory में किसी location का नाम होता है।



Computer Memory Space

मान लीजिए ये आपका computer memory space है। आप इसमें कुछ data store करवाना चाहते हैं। जैसे की किसी व्यक्ति की उम्र आदि। Computer की memory में data store करवाने से पहले आप बताते ही की आप किस तरह का data store करेंगे। ये आप data types के द्वारा define करते हैं। उसी के according आपको memory में space मिलता है। यानि की यदि आपने int define किया है तो 2 bytes आपको memory में compiler allot करेगा।

इसके साथ ही आपको उस memory location का नाम भी define करना होता है। ताकि आप जब भी चाहे उस memory location में store की गई value को इस नाम के द्वारा access कर सके। इस नाम को ही variable कहते हैं। Variables की values changeable होती है। आप एक value को हटाकर दूसरी value डाल सकते हैं। ऐसा आप manually खुद भी कर सकते हैं या फिर dynamically (program execution के दौरान) भी कर सकते हैं।



Google द्वारा विज्ञापन बंद कर दिया गया

यह विज्ञापन देखना बंद करें

Google द्वारा विज्ञापन ①

Creating variables in C

Variables create करने के लिए सबसे पहले आप data type define करते हैं। इसके बाद आप एक unique name define करते हैं। इसका structure इस प्रकार होता है।

```
dataType variableName; // without assigning value

dataType variableName = value; // with value assignment
```

उदाहरण के लिए निचे दिए गए statement को देखिये।

```
int Age = 25;
```

इस statement के द्वारा एक integer variable create किया गया है, जिसका नाम Age है और इस variable को 25 value assign की गई है। आइये अब समझते हैं की compiler इस statement को किस प्रकार interpret करता है।

जब compiler सबसे पहले int को interpret करता है तो वह computer की memory में से 2 bytes की memory allot करता है। इसके बाद जब compiler Age को interpret करता है तो वह उस 2 bytes की memory को age नाम दे देता है। इसके बाद जब compiler = 25 को interpret करता है तो 25 को इस memory location store कर देता है।



अब जब भी आप इस value को access करना चाहते हैं Age के द्वारा इसे access कर सकते हैं।

Scope of variables

कोई variable program में कहाँ तक काम कर सकता है। ये उसका scope होता है। Scope के according variables को 2 categories में divide किया गया है।

Local variables

Local variables वो variables होते हैं जो program के किसी छोटे block में define किये जाते हैं जैसे की function या control statement blocks या कोई और block ({ }) भी हो सकता है। इस तरह के variables का यूज़ सिर्फ इस block तक ही limited रहता है। जैसे की यदि आपने किसी



function में कोई variable create किया है तो आप उस variable को उस function के बाहर access नहीं कर सकते है। उदाहरण के लिए निचे दिए गए program को देखिये।

```
#include <stdio.h>

void myFunction();

int main()
{
    int num=6;
    myFunction();
    printf("Num in main() : %d", num);

    return 0;
}

void myFunction()
{
    int num= 5;
    printf("Num in myFunction : %d\n",num);
}
```

ऊपर दिया गया program निचे दिया गया output generate करता है।

```
Num in myFuntion() : 5
Num in main() : 6
```

Global variables

Global variables वो variables होते है जिनका scope पुरे program में होता है। इन variables को आप पुरे program में कंही भी access कर सकते है। इसका उदाहरण निचे दिया जा रहा है।

```
#include <stdio.h>

int num=5;

void myFunction();

int main()
{
    myFunction();
    printf("Num in main() : %d",num);
    return 0;
}
```



```
void myFunction()
{
    printf("Num in myFunction : %d\n",num);
}
```

ऊपर दिया गया program निचे दिया गया output generate करता है।

```
Num in myFunction() : 5
Num in main() : 5
```

Constants in C

Constants वो variables होते हैं जिनकी value change नहीं होती है। जब भी आप कोई constant declare करते हैं तो program के execution के दौरान उसकी value fixed रहती है। यदि इसकी value change करने की कोशिश की जाती है तो program में error आ जाती है। इन्हें literals भी कहा जाता है।

Constants को आप 2 तरह से declare कर सकते हैं।

- Using #define
- Using Const keyword

Using #define

#define एक pre-processor है इसे यूज करके आप constant declare कर सकते हैं। इसका उदाहरण निचे दिया जा रहा है।

```
# include <stdio.h>

#define result 10

int main()
{
    int a=5, b=6;

    result = a + b; /* WRONG, (ERROR) Value of constant result variable
can not be changed. */

    printf("%d", result);

    return 0;
}
```

ऊपर दिया गया program निचे दिया गया output generate करता है।



```
error : lvalue required as left operand of assignment
      result = a + b;
                ^
```

Using const keyword

Const keyword के द्वारा भी आप constants declare कर सकते हैं। इसका उदाहरण निचे दिया जा रहा है।

```
#include <stdio.h>

int main()
{

const int a=5;

const int b=6;

int c;

c = a+b;

printf("Result is : %d",a);

return 0;

}
```

ऊपर दिया गया program निचे दिया गया output generate करता है।

```
Result is : 11
```

© Best Hindi Tutorials 7/20/2016         

4  Replies so far - Add your comment

Arvind Patil

शनिवार, 01 अक्टूबर, 2016

why use return 0;
उत्तर दें

उत्तर

vipin sharma

शनिवार, 01 अक्टूबर, 2016

Jab bhi aap kisi function ka return type VOID define na karke INT define karte h to iska mtlb hota h ki aapka function INT return karega, lekin fir bhi yedi function koi value return nahi karta h to esi situation me aap 0 return karvate h, Kyonki 0 bhi integer hota h or isse output par koi asar bhi nahi padta h, Yedi aap kuch bhi return nahi karvate h to program error show karta h.



उत्तर दें

ratnesh jawre

सोमवार, 05 दिसंबर, 2016

ham return karwate kyon h iski jarurat kya ????

उत्तर दें

उत्तर

Vipin Sharma

बुधवार, 07 दिसंबर, 2016

Basically hota kya h ki ek function Input->Process->Output ke model ko follow karta h. Mtlb ye hua ki aap function me input denge or function aapko output dega.

Udaharan ke liye yedi aap function me 2 number read kar rahe h jo ki integer h to iska mtlb ye hua ki aapke function ka result ya fir output jo hoga vo bhi integer number hoga. Isliye aap function create karte samay return type define karte h.

Or isi model ki wajah se aap function se ek value return karvate h. kyonki yedi aap koi result ya output nahi dete h to function create karane ka koi sense nahi h.

yedi me addition ke function 2 or 4 pass karu or vo mujhe add karke kuch bhi return na kare to sochiye kesa hoga?

or yedi kabhi kisi function me aap kuch return nahi karna chahte h to aap return 0; statement bhi use kar sakte h.

उत्तर दें

अपनी टिप्पणी लिखें...

इस रूप में टिप्पणी करें: Dr. Namita Mitt ▼

प्रस्थान करें

प्रकाशित करें

पूर्वावलोकन

मुझे सूचित करें



Best Hindi Tutorials

Computer Science and IT tutorials in Hindi

[Home](#) [About](#) [Contact](#) [Recommend](#)

Start Download - PDF Now

Convert From Doc to PDF, PDF to Doc Simply With The Free On-line App!
fromdoctopdf.com पर जाएं



C Language

What is Programming?

C - Introduction

C - Tokens, Identifiers & Keywords

C - Data Types

C - Variables

C - First Program

C - Flow Chart

C - Operators

C - Decision Making

C - Looping

C - Arrays

C - Strings

C - Functions

C - Structures

C - Unions

C - Recursion

C - Pointers

C - Type Casting

C - Storage Classes

C - Console I/O

C - File Handling

C - Error Handling

C - Header Files

C - Preprocessor Directives

C Arrays

- Introduction to **C arrays in Hindi**
- Declaring *C arrays in Hindi*
- Initializing C arrays in Hindi
- Accessing elements of C arrays in Hindi

Introduction to C arrays

मान लीजिये आप एक ऐसा प्रोग्राम बना रहे हैं जो employees का नाम computer में store करता है। अब मान लीजिये आपकी company में 200 employees हैं। आप इन 200 employees के नाम किस तरह से store करेंगे। यदि आप सोच रहे हैं की आप 200 variables create करेंगे तो ये एक बहुत ही complex approach होगी।

इसमें आपको program बनाने में बहुत समय लग जायेगा और program भी हद से ज्यादा बड़ा हो जायेगा। ये आपके time और computer memory space दोनों का wastage है। और साथ ही आप इतने सारे variables के नाम सोच भी नहीं सकते हैं और यदि सोच भी लेंगे तो program में यूज करने के लिए उन्हें याद तो definitely नहीं रख सकते हैं।

मेरे पास आपके लिए इससे भी better approach है और उस approach को C language में array कहते हैं। Array similar type की values का collection होता है। Similar type से यँहा मेरा मतलब similar data type से है जैसे की int, float, Char आदि।

C आपको arrays के द्वारा ऐसी facility provide करती है की आप सिर्फ एक variable create करे और उस variable में 200 employees के नाम (या जो भी information आप store करना चाहते हैं) store कर ले। आप सोच रहे होंगे की एक variable में इतने सारे नाम कैसे store करेंगे। इसके बारे में मुझे आपको बताने में बहुत खुशी होगी। लेकिन उससे पहले आइये देखते हैं की array को create कैसे करते हैं।

Creating c arrays

Array एक structured data type होता है। जैसा की आपने C Structures की tutorial में पढ़ा होगा की एक structure में आप कई variables create कर सकते हैं। यँहा भी situation कुछ कुछ





वैसी ही है। जब भी आप एक array create करते हैं तो आप array का नाम और आप उसमें कितनी values store करने वाले हैं ये define करते हैं।

जैसे की मान लीजिये आप 5 numbers store करना चाहते हैं तो उसके लिए आप array create कर सकते हैं। C में arrays create करने का general structure नीचे दिया गया है।

```
data_type array-name[size];
```

size से आप define करते हैं की आप कितनी values store करना चाहते हैं। इसका उदाहरण नीचे दिया जा रहा है।

```
int num[5];
```

उपर दिए गए उदाहरण में array का नाम num है और उसमें आप कोई भी 5 integer values store कर सकते हैं।

Initializing C arrays

यँहा तक मैंने आपको बताया है की array कैसे create करते हैं। अब आपके सबसे महत्वपूर्ण question पर आते हैं की आप इस array में 5 values कैसे store करेंगे। तो मैं आपको बता दूँ की जब भी आप array create करते हैं तो जितनी उसकी size होती है उतनी ही locations उस array को memory में allocate हो जाती है। और वो locations उतने ही index numbers के नाम से allocate होती है।

Index numbers हर location का एक unique नाम होता है। जैसे की यँहा ऊपर दिए उदाहरण में 5 index numbers होंगे। आपको एक बात हमेशा याद रखनी चाहिए की array की index हमेशा zero से शुरू होती है।

```
num[0]
num[1]
num[2]
num[3]
num[4]
```

Array के नाम और index number से आप create की गयी हर location में value store करवा सकते हैं और बाद में उससे value access भी कर सकते हैं।

(Note : Arrays की index हमेशा zero से शुरू होती है।)

जैसे की ऊपर create किये गए array में आप इस प्रकार value insert करवा सकते हैं।

```
num[0] = 50;
num[1] = 100;
num[2] = 150;
num[3] = 200;
```



```
num[4] = 250;
```

आप चाहे तो हर location को अलग से value assign करने की बजाय एक साथ भी सभी values को assign कर सकते हैं। ऐसा आप इस प्रकार कर सकते हैं।

```
int num[5] = {50,100,150,200,250};
```

और यदि आप values यूजर से run time में store करवाना चाहते हैं तो ऐसा आप इस प्रकार कर सकते हैं।

```
for(int i=0; i<=5; i++)  
{  
    scanf("%d",&num[i]);  
}
```

Accessing array elements

अब तक मैंने आपको arrays create करना और उनमें values store करना बताया है। इस section में मैं आपको उन values को access करना बताऊंगा। Array elements को access करके आप उन पर वो सभी operations perform कर सकते हैं जो आप normal variables के साथ perform कर सकते हैं। उदाहरण के लिए आप 2 array elements को add करवा सकते हैं।

जहाँ भी array element को access करना चाहते हैं उस जगह आप array का नाम और index number लिखते हैं। उदाहरण के लिए निचे लिखे गए statement को देखिये।

```
num[2] = num[0] + num [1];
```

ऊपर दिए गए statement में शुरू की 2 values को add करके 3 location पर store करवाया गया है।

यदि आप किसी एक value को print करवाना चाहते हैं तो ऐसा आप इस प्रकार कर सकते हैं।

```
printf("%d",num[3]);
```

ऊपर दिया गया statement 4th location की value print करेगा जो की यँहा दिए गए example में 200 है।

यदि आप पूरे array को एक साथ print करना चाहते हैं तो ऐसा आप loop की मदद से कर सकते हैं। ये ध्यान रखे की loop उतना ही चले जितनी की values आपके array में है। इसका उदाहरण नीचे दिया जा रहा है।

```
for(int i=0; i<=5;i++)  
{
```



```
printf("%d",num[i]);  
}
```

Example

```
#include <stdio.h>  
#include <conio.h>  
  
int main()  
{  
    int arr[5];  
    int i;  
  
    printf("Please enter 5 array elements : \n");  
    for(i=0;i<=4;i++)  
    {  
        scanf("%d",arr[i]);  
    }  
  
    printf("Array is:");  
    for(i=0;i<=4;i++)  
    {  
        printf("%d\t",arr[i]);  
    }  
  
    return 0;  
}
```

उपर दिया गया program निचे दिया गया output generate करता है।

```
Please enter 5 array elements : 5 4 3 2 1  
Array is : 5 4 3 2 1
```

Two Dimensional Arrays

एक normal array में data list की form में store किया जाता है जिसमें एक के बाद दूसरा element होता है। यदि आप data tabular form में store करना चाहते हैं तो इसके लिए आप two dimensional array create कर सकते हैं। उदाहरण के लिए आप 4 employees की id और उनका phone number store करना चाहते हैं। इसके लिए table इस प्रकार बनेगी।

101	7599393
102	9384940
103	9458940
104	9129399

इस प्रकार की table आप two dimensional array के द्वारा memory में create कर सकते हैं।

Creating Two Dimensional Array

C language में two dimensional array create करने का general syntax निचे दिया जा रहा है।

```
data-type array_name[row][column];
```

Two dimensional array को row और column के संदर्भ में define किया जाता है। सबसे पहले आप array का type define करते हैं। इसके बाद आप array का एक unique नाम define करते हैं। इसके बाद आप जितनी row create करना चाहते हैं उतनी संख्या angular bracket में define करेंगे।



इसके बाद आप जितने columns create करना चाहते हैं उतनी संख्या दूसरे angular bracket में define करेंगे।

उदाहरण के लिए ऊपर दी गयी table आप two dimensional array के माध्यम से memory में इस प्रकार create करेंगे।

```
int myArray[4][2];
```

ऊपर दिया गया statement memory में एक table (two dimensional array) create करेगा जिसकी 4 rows और 2 columns होंगे।

Initializing Two Dimensional Array

Two dimensional array को आप किसी variable की तरह initialize कर सकते हैं। जब आप इस प्रकार initialize करते हैं तो हर position के लिए separately value assign करते हैं। उदाहरण के लिए आप पहली row के पहले column में value store करना चाहते हैं तो ऐसा आप इस प्रकार करेंगे।

```
myArray[1][1]=101;
```

इसी प्रकार second column में phone number आप इस प्रकार store कर सकते हैं।

```
myArray[1][2]=7599393;
```

यदि आप user से इस array में value input करवाना चाहते हैं तो इसके लिए आप 2 loops use करेंगे। पहला loop rows को iterate करेगा और इसे row की संख्या तक ही चलाया जायेगा। दूसरा loop columns को iterate करेगा और इसे भी columns की संख्या तक ही चलाया जायेगा। इसका उदाहरण निचे दिया जा रहा है।

```
for(int i=0;i<4;i++)
{
    for(int j=0;j<2;j++)
    {
        scanf("%d",myArray[i][j]);
    }
}
```

Accessing Two Dimensional Array

Two dimensional array में से यदि आप किसी single element को access करना चाहते हैं तो ऐसा आप उसकी position index को access करके कर सकते हैं। उदाहरण के लिए यदि आप 2nd row के 1st column की value print करना चाहते हैं तो इसके लिए आप इस प्रकार statement लिखेंगे।

```
printf("Element at 2nd row 1st column is :%d",myArray[2][1]);
```

यदि आप array के सभी elements को एक बार में print करना चाहते हैं तो इसके लिए आप 2 loops use करते हैं। इसका उदाहरण निचे दिया जा रहा है।

```
for(int i=0;i<4;i++)
{
    for(int j=0;j<2;j++)
```



```
    {  
        printf("%d",myArray[i][j]);  
    }  
}
```

Example

```
#include <stdio.h>  
  
int main()  
{  
    int myArray[4][2];  
    int i,j;  
  
    printf("Please enter array elements : ")  
    for(i=0;i<4;i++)  
    {  
        for(j=0;j<2;j++)  
        {  
            scanf("%d",&myArray[i][j]);  
        }  
    }  
  
    printf("Arrays is : \n")  
    for(i=0;i<4;i++)  
    {  
        for(j=0;j<2;j++)  
        {  
            printf("%d",myArray[i][j]);  
        }  
        printf("\n");  
    }  
}
```

ऊपर दिया program निचे दिया गया output generate करता है।

```
Please enter array elements : 1 2 3 4 5 6 7 8  
Arrays is :  
1 2  
3 4  
5 6  
7 8
```

© Best Hindi Tutorials 8/23/2016          

 Leave a comment

अपनी टिप्पणी लिखें...

इस रूप में टिप्पणी करें: Dr. Namita Mitt ▼

प्रस्थान करें

प्रकाशित करें

पूर्वावलोकन

मुझे सूचित करें



Best Hindi Tutorials

Computer Science and IT tutorials in Hindi

[Home](#) [About](#) [Contact](#) [Recommend](#)



Build scalable
Machine Learning models

[TRY IT FREE](#)

C Language

What is Programming?

C - Introduction

C - Tokens, Identifiers & Keywords

C - Data Types

C - Variables

C - First Program

C - Flow Chart

C - Operators

C - Decision Making

C - Looping

C - Arrays

C - Strings

C - Functions

C - Structures

C - Unions

C - Recursion

C - Pointers

C - Type Casting

C - Storage Classes

C - Console I/O

C - File Handling

C - Error Handling

C - Header Files

C - Preprocessor Directives

C Looping

- Introduction to **C language looping in Hindi**
- C language *while loop in Hindi*
- C language do-while loop in Hindi
- C language for loop in Hindi

Introduction to C language looping

यदि मैं आपसे कहूँ की आपको C language in Hindi 10 बार print करवाना है तो ऐसा आप किस तरह करेंगे? सामान्यतया: ऐसा करने के लिए आप 10 printf() statements लिखेंगे। ये एक बेकार approach है। ऐसा करने से programmer का time और computer की memory waste होती है।

C language में आपको loops provide किये गए हैं। Loops की मदद से आप एक ही statement को बार बार execute करवा सकते हैं। हर तरह का loop एक block provide करता है जिसमें वो statements लिखे जाते हैं जिन्हें आप एक से ज्यादा बार execute करवाना चाहते हैं।

Loop 3 चीजों से मिलकर बना होता है।

- Initial variable - ये वो variable होता है जहाँ से आप loop को start करते हैं। ये एक integer variable होता है। इस variable को तब तक increment किया जाता है जब तक की दी गयी condition false ना हो जाये। इस variable को loop की condition में include किया जाता है।
- Condition - ये वो condition होती है जो loop को control करती है। जब तक ये condition true रहती है loop execute होता रहता है। जैसे ही ये condition false होती है loop terminate हो जाता है।
- Increment - आप कितने number से या कैसे initial variable/condition को increment करना चाहते हैं ये increment part में define किया जाता है।

C language आपको 3 तरह के loops प्रोवाइड करती है।

- While loop
- Do-while loop
- For loop



Google द्वारा विज्ञापन बंद कर दिया गया

यह विज्ञापन देखना बंद करें

Google द्वारा विज्ञापन ①

आइये अब C language में use होने वाले इन loops के बारे में detail से जानने का प्रयास करते हैं।

C while loop

While एक simple loop होता है ये जब तक condition true रहती है तब तक execute होता है। Condition के false होने पर ये loop terminate हो जाता है। While loop का general syntax नीचे दिया जा रहा है।

```
initial variable declaration;

while(condition)
{
    //statements
    //increment
}
```

Condition यदि पहली बार में ही false हो तो compiler loop में enter ही नहीं होता है। Loop को totally skip कर दिया जाता है। आइये while लूप को एक उदाहरण से समझने का प्रयास करते हैं।

```
#include <stdio.h>

void main()
{
    int num=1;

    while(num<=10)
    {
        printf("%d\t",num);
        num++;
    }
}
```

ऊपर दिए गए उदाहरण में पहले num variable को 1 value के साथ set किया गया है। Loop के अंदर condition दी गयी है की जब तक num 10 से कम या 10 के बराबर ना हो loop execute होता रहे। Loop के अंदर हर iteration में num variable की value print की जा रही है। इसके बाद num variable को increment किया जा रहा है। ये program निचे दिया गया output generate करता है।

```
1 2 3 4 5 6 7 8 9 10
```

यदि initial variable को increment ना किया जाये तो condition कभी false नहीं होगी। ऐसे में loop infinite time तक चलता जायेगा।



C Do-while loop

Do while loop भी while loop की तरह ही होता है। लेकिन इस loop में पहले condition check होने के बजाय पहले statements execute होते हैं उसके बाद में condition check होती है। लेकिन ऐसा सिर्फ first time होता है। Do while loop का general syntax नीचे दिया जा रहा है।

```
do
{
    //statements;
    //increment;
}while(condition);
```

जैसे ही compiler do section में आता है तो do block के सभी statements execute कर दिए जाते हैं और initial variable को increment कर दिया जाता है। इसके बाद compiler while condition को check करता है। यदि condition true होती है तो do block को वापस execute किया जाता है। लेकिन यदि condition false हो तो loop terminate करके compiler आगे बढ़ जाता है।

While की condition चाहे true हो या false do block के statements एक बार जरूर execute होते हैं। आइये do while loop को एक उदाहरण से समझने का प्रयास करते हैं।

```
#include <stdio.h>

void main()
{
    int num=1;

    do
    {
        printf("%d\n",num);
        num++;
    }while(num>2);

    printf("Num is not greater than 2 that is why loop terminated!!");
}
```

उपर दिए गए उदाहरण में हालाँकि condition false है लेकिन फिर भी do block में दिया गया statement एक बार जरूर execute होगा। ये program निचे दिया गया output generate करता है।

```
1
Num is not greater than 2 that is why loop terminated!!
```

C For loop



C loops में for loop सबसे ज्यादा use किया जाता है। ये loop बहुत ही easy होता है और एक single statement में define हो जाता है। इस loop का general structure नीचे दिया जा रहा है।

```
for(initialvariable;condition;increment)
{
    //statements
}
```

For loop के बारे में खास बात ये है की आप एक statement में तीनों elements define किये जाते हैं। इस loop को नीचे उदाहरण द्वारा समझाया गया है।

```
#include <stdio.h>

void main()
{
    int num;
    for(num=1;num<=10;num++)
    {
        printf("%d",num);
    }
}
```

ऊपर दिया गया program निचे दिया गया output generate करता है।

```
1 2 3 4 5 6 7 8 9
```

© Best Hindi Tutorials 9/15/2016



2 Replies so far - Add your comment

Rajat Kumar

सोमवार, 13 मार्च, 2017

What is the difference between 'int main()', 'void main()' and 'main()' function in the C programming Hindi me Explain kigiye

उत्तर दें

उत्तर

Best Hindi Tutorials

सोमवार, 13 मार्च, 2017

int main() ka matlab h ki main function ek integer value return karega. yedi aap function ke ant me koi phi value return nahi karate h to error generate hoti h, Error se baccon ke like esi situation me aap 0 return karate h, ye main() function ko define Karne ka standard tarika h.

void main() ka matlab h ki function koi bhi value return nahi karega. jab aap is prakar main() function ko define karte h to koi phi value return nahi karvane par phi error generate nahi hoti h.

main() ek function h, ise is tarah aap use nahi Kar sakte h. kisi bhi function ka return type define Karna aavshyak Hota h.



उत्तर दें

अपनी टिप्पणी लिखें...

इस रूप में टिप्पणी करें: Google खाता ▼

प्रकाशित करें

पूर्वावलोकन



Unit I-

Introduction to 'C' Language

1. Introduction to Program steps
Overview of Programming Languages, Compilation, Linking and Loading, Testing and Debugging, Documentation.

Unit II-

2. Introduction to 'C' Language
Character set, Variables and identifiers, Built-in Data Types, Variable Definition, Arithmetic operators and Expressions, constants and Literals, Simple assignment statement, Basic input/output statement, Basic input/output statement, Simple 'C' programs.

sem
IV

S-II