सी लैंग्वेज क्या है? (What is C Language in Hindi)

C Language एक **general-purpose** प्रोग्रामिंग लैंग्वेज है जिसको **Dennis Ritchie** ने 1972 में **AT & T'S Bell Telephone Laboratories** में बनाया था | Dennis Ritchie एक operating system बनाना चाहते थे जिसका नाम था **Unix Operating System**, इसको बनाने के लिए ही Dennis Ritchie ने सी लैंग्वेज Developed किया था |

खास बात ये है कि सी लैंग्वेज की मदद से हम **low level** प्रोग्रामिंग कर सकते है | इसके इस Feature के कारण C programming language का उपयोग System software जैसे – **Operating system**, **Device Driver**, **Compiler** आदि बनाने के लिए किया जाता है |

C Language में बाकि सभी प्रोग्रामिंग लैंग्वेज के बेसिक फीचर्स कवर हो जाते है (जैसे – **Variable**, **Data Types**, **Array**, **String**, **Function**, **Structure**, **Pointer**, **Loop** आदि) जिसके कारण C language को बाकि सभी प्रोग्रामिंग लैंग्वेज का **Mother Language** कहा जाता है |

सी प्रोग्राम की मूल संरचना (Basic Structure of C Program in Hindi)

सी लैंग्वेज में जब भी हम कोई प्रोग्राम बनाते है तो उस प्रोग्राम को हम छः डिफरेंट section में बाँट सकते है ये सेक्शन कुछ इस प्रकार है -:

1. Documentation (Documentation Section)
2. Preprocessor Statements (Link Section)
3. Definition Section
4. Global Declarations Section
5. Main functions Section
6. User-Defined Functions or Sub Program Section

सी लैंग्वेज में ये सभी छः सेक्शन मिलकर C Program का Basic Structure बनाते है आइये अब हम इन सभी Section के बारे में विस्तार से जानते है -:

1. Documentation (Documentation Section)

प्रोग्राम को describe करने के लिए प्रोग्रामर Documentation Section में Comments लिखता है | Comments को कम्पाइलर ignore कर देता है, उसे स्क्रीन में प्रिंट नहीं करता | Comments सिर्फ उस प्रोग्राम को describe करने के काम में आता है |

प्रोग्रामर Comments के अंदर उस प्रोग्राम का नाम, Author Name जो उस प्रोग्राम को बना रहा है और दूसरी जानकारियाँ जैसे – प्रोग्राम का date , उसका उद्देश्य आदि | ये सब कुछ Documentation Section के अंतर्गत लिखा जाता है |

2. Preprocessor Statements (Link Section)

Link Section के अंदर हम अपने प्रोग्राम में उपयोग होने वाले सभी Header Files को Declare करते है | Link Section से हम Compiler को इंस्ट्रक्शन देते है कि वो system libraries से उन Header Files को जिसे हमने लिंक सेक्शन में डिक्लेअर किया है उसे हमारे इस प्रोग्राम में लिंक दे |

Example -:

C Language

```c
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<math.h>
```

Link Section में इन सब Header Files के आलावा बहुत सारे Header Files भी होते है जिसे हम जरुरत पड़ने पर अपने प्रोग्राम में लिंक कर सकते है |

3. Definition Section

सी लैंग्वेज में हम जितने Symbolic Constant का उपयोग करते है उसका Definition इस Section में करते है इसलिए इस सेक्शन को Definition Section कहते है |

Macros का Definition भी इसी Section में किया जाता है |

Example -:

C

```c
#define PI 3.14
```

4. Global Declarations Section

Global Declarations Section सेक्शन के अंदर हम ऐसे वेरिएबल को Declare करते है जिनको हम अपने प्रोग्राम में कही भी उपयोग करना चाहते है, ऐसे वेरिएबल ग्लोबल वेरिएबल कहलाते है इन वेरिएबल्स को हम किसी भी फंक्शन में कही पर भी उपयोग कर सकते है |

Global Declarations Section में ही हम ऐसे फंक्शन को भी Declare करते है जिनको हम अपने प्रोग्राम में कही भी उपयोग चाहते है और ऐसे फंक्शन Global Function कहलाते है |

Example -:

```c
int area(int x);  // Global Function
int n;            // Global Variable
void main()
{
 statements;
.
.
}
```

5. Main functions Section

सी लैंग्वेज में जब भी हम कोई प्रोग्राम बनाते है तो उस प्रोग्राम में एक main() फंक्शन होता ही है | main() फंक्शन curly brackets से स्टार्ट होता है और curly brackets से ही ख़तम होता है | main() फंक्शन में हम इन curly brackets के अंदर ही अपना Statements लिखते है |

main() फंक्शन के अंदर हम जो कोड लिखते है वो दो पार्ट में होते है एक Declaration Part और दूसरा Execution Part | Declaration Part में हम ऐसे वेरिएबल्स को डिक्लेअर करते है जिनका उपयोग हमे Execution Part में करना होता है आइये इसको हम एक Example से समझते है -:

Example -:

```c
int main(void)
{
 int n=15;              // Declaration Part
 printf(" n = %d ",n);  // Execution Part
 return (0);
}
```

6. User-Defined Functions or Sub Program Section

इस सेक्शन के अंदर सभी User-Defined Functions को Declare किया जाता है |

Example -:

```
int sum( int x, int y)
{
  return x+y ;
}
```

वेरिएबल क्या है? (What Is Variable In C In Hindi)

वेरिएबल मेमोरी में उस स्थान का नाम है जिस स्थान पर हम कोई डाटा या इनफार्मेशन स्टोर करके रखते है | एक प्रोग्राम के अंदर बने वेरिएबल को हम उस प्रोग्राम के Execution के दौरान कई बार बदल और अपडेट कर सकते है |

वेरिएबल एक तरह का Identifiers है जिसका उपयोग हम डाटा को स्टोर करके रखने के लिए करते है |

आइये वेरिएबल को हम इस एक उदहारण के द्वारा और अच्छे से समझते है -:

जब भी हमे कोई इनफार्मेशन या डाटा अपने कम्प्यूटर में या फिर लैपटॉप में स्टोर करना होता है तब उसे हम अपने कंप्यूटर के मेमोरी में स्टोर करके रखते है | मेमोरी में जिस स्थान पर वो डाटा रखते है उसे ऑपरेटिंग सिस्टम एक कॉम्प्लेक्स Address प्रदान करता है जिसे हम जैसे नार्मल यूजर के लिए याद रखना मुश्किल होता है |

ऐसे में ऑपरेटिंग सिस्टम नार्मल यूजर को ये फीचर भी प्रदान करता है की वो यूजर उस डाटा को जिस स्थान में रख रहा है उस स्थान का नाम वो खुद से तय कर सकता है और चाहे तो वो कोई फोल्डर बना कर उस फोल्डर का नाम रखकर उस फोल्डर के अंदर उस डाटा या इनफार्मेशन को रख सकता है जिससे अगर बाद में, उसे उस डाटा या इनफार्मेशन की जरुरत पड़ी तो वह उसे आसानी से ढूंढ और प्राप्त कर सकते है |

आसान शब्दो में कहूं तो **वेरिएबल मेमोरी में उस स्थान का नाम है जिस स्थान पर हम कोई डाटा या इनफार्मेशन स्टोर करके रखते है** जिससे आगे अगर उस इनफार्मेशन का काम होने पर आसानी से ढूंढ और प्राप्त कर उसका उपयोग कर सके |

Syntax to declare a variable

Data_type Variable_name

The example of declaring the variable is given below:

```c
char my_name = "Jeetu Sahu";
```

इस Example में char एक Data type है और my_name वेरिएबल का नाम है जिसमे Jeetu Sahu नाम का डाटा Store हो रहा है | हम char डाटा टाइप के आलावा और भी दूसरे डाटा टाइप का उपयोग करके किसी भी तरह के Data को Store कर सकते है Data type के बारे में और ज्यादा जानने के लिए यहाँ से पढ़ने Datatype क्या है ?

वेरिएबल का नाम Declare करने के नियम – Rules for Naming a Variable

1. वेरिएबल के नाम में अल्फाबेट, नंबर, और अंडरस्कोर का उपयोग ही हो सकता है |
2. वेरिएबल के नाम का पहला अक्षर या तो कोई लेटर होगा या फिर अंडरस्कोर होगा |
3. वेरिएबल का नाम नंबर से शुरू नहीं हो सकता |
4. वेरिएबल का नाम मीनिंगफुल होना चाहिए |
5. ब्लैंक और खाली स्पेस का उपयोग वेरिएबल के नाम में नहीं किया जा सकता |
6. किसी कीवर्ड का उपयोग वेरिएबल के नाम के रूप में नहीं किया जा सकता |
7. सी लैंग्वेज में अगर आप किसी वेरिएबल को लोअरकेस लेटर में और उसी वेरिएबल को अपरकेस लेटर में लिखने पर ये दोनों सी लैंग्वेज की नजर में अलग अलग वेरिएबल है क्योकि सी लैंग्वेज एक केस सेंसेटिव लैंग्वेज है |

वेरिएबल के प्रकार (Types of Variables in C)

सी लैंग्वेज में वेरिएबल निम्नलिखित प्रकार के होते है -:

1. Local variable
2. Global variable
3. Static variable
4. Automatic variable
5. External variable

1. Local variable in C

ऐसा वेरिएबल जिसे हम किसी फंक्शन या ब्लॉक के अंदर Declare करते है, **local variable** कहलाते है | Local variable का स्कोप केवल उस ब्लॉक या फंक्शन तक ही होता है ऐसे वेरिएबल को हम किसी दूसरे फंक्शन में उपयोग नहीं कर सकते |

ऐसे वेरिएबल का डिक्लेरेशन शुरुआत में ही करना पड़ता है और लोकल वेरिएबल को उपयोग करने से पहले initialize करने की जरुरत पड़ती है |

2. Global variable in C

ग्लोबल वेरिएबल ऐसे वेरिएबल होते है जो फंक्शन या ब्लॉक के बाहर डिक्लेअर होते है | ग्लोबल वेरिएबल को हम किसी दूसरे ब्लॉक या फंक्शन में उपयोग कर सकते | इस वेरिएबल की वैल्यू को कोई भी function में उपयोग और चेंज किया जा सकता है |

इस वेरिएबल को उपयोग करने से पहले डिक्लेअर करना आवश्यक होता है |

3. Static variable in C

Static वेरिएबल डिक्लेयर करने के लिए Static कीवर्ड का उपयोग किया जाता है | Static वेरिएबल को यदि हम किसी फंक्शन या ब्लॉक के अंदर डिक्लेयर करते है तो इसे local Static variable कहते है और यदि Static variable का डिक्लेरेशन फंक्शन के बाहर करते है तो इसे global Static variable कहते है |

Static variable की वैल्यू वैल्यू डिक्लेअर करते टाइम by default जीरो होता है जबकि नार्मल वेरिएबल में by default गार्बेज वैल्यू होता है |

4. External variable in C

External variable को हम extern कीवर्ड का उपयोग करके डिक्लेअर करते है | External variable का उपयोग हम विभिन्न source files में कर सकते है |

5. Automatic variable in C

सी लैंग्वेज में हम जितने भी वेरिएबल ब्लॉक के अंदर डिक्लेअर करते है वो सभी Automatic variable होते है | इस वेरिएबल को हमें auto keyword का उपयोग करके डिक्लेअर करना बहुत ज्यादा जरूरी नहीं होता ये एक नार्मल वेरिएबल जैसा ही होता है या कहे कि सभी नार्मल वेरिएबल by डिफ़ॉल्ट Automatic variable होते है |

Data Types क्या है (What is Data Types in C in Hindi)

सी लैंग्वेज में जब भी हम किसी डेटा या इन्फॉर्मेशन स्टोर करने के लिए कोई वेरिएबल बनाते है तब हमे उस वेरिएबल को डिक्लेअर करते समय ये भी डिक्लेअर करना पड़ता है कि वो वेरिएबल किस टाइप का डेटा स्टोर करने वाला है |

ये डेटा टाइप **int**, **char**, **float**, **double** कुछ भी हो सकते है इन डेटा टाइप को देख कर पता चलता है कि वेरिएबल में किस टाइप की वैल्यू स्टोर होने वाला है |

जैसे की **int** डेटा टाइप से बने वेरिएबल में हम **Integer** वैल्यू को स्टोर करते है, **char** डेटा टाइप से बने वेरिएबल में हम **Character** टाइप के डेटा को स्टोर करते है तथा **Float** डेटा टाइप के द्वारा बने वेरिएबल में हम **फ्लोटिंग पॉइंट** वाले वैल्यू को स्टोर करते है |

आइये डेटा टाइप्स को हम एक example से समझते है -: **Example of Data Types in C**

```c
//Example of Data Types in C
#include<stdio.h>
int main()
{
   int   x  = 14;
   char  ch = 'a';
   float f1 = 25.7;
   printf(" x = %d, ch = %c, f1 = %f ",x,ch,f1);
   return (0);
}
```

Data Types के प्रकार (Types of Data Types In C Language)

सी लैंग्वेज में मुख्यतः तीन तरह के Data Types होते है -:

1. Pre-defined data types
2. Derived Data Types
3. User-defined data type

1. Pre-defined Data Types

ऐसे डेटा टाइप्स **Pre-defined Data Types** कहलाते है जो पहले से Defined होते है तथा जिनको अलग से और Defined करने की जरुरत नहीं होती जैसे कि – int , char, float, double, void आदि |

ये डेटा टाइप्स **Basic Data Types** में आते है | इन **Data Types** का मतलब कम्पाइलर को अलग से बताना नहीं पड़ता क्योकि ये पहले से Defined होते है और ऐसे Data Types जो Keywords भी है उन डेटा टाइप्स को **Primitive data types** कहते है |

List of Predefined Data Types

Pre-defined Data Types	**Examples**
Integer Type	int, long int, short int, unsigned int
Character Type	char
Floating Point Data Type	Float , double

Integer Type

- "int" कीवर्ड का उपयोग इन्टिजर टाइप के वेरिएबल को बनाने के लिए किया जाता है |
- Integer Type के वेरिएबल में हम न्यूमेरिक वैल्यू स्टोर करते है |
- "int" डेटा टाइप से बना वेरिएबल 2 byte , 4 byte और 8 byte डेटा स्टोर कर सकता है | मगर ये चीज कम्पाइलर पर डिपेंड करता है |
- कुछ कम्पाइलर int डेटा टाइप से बने वेरिएबल में 2 byte डेटा स्टोर करता है तो कुछ 4 byte डेटा स्टोर करता है |
- int (2 byte), -32,768 से +32,767 तक के वैल्यू को स्टोर कर सकता है |
- int (4 byte), -2,147,483,648 से +2,147,483,647.तक के वैल्यू को स्टोर कर सकता है |
- अगर आपको इससे ज्यादा वैल्यू का डेटा वेरिएबल में स्टोर करना है तो आप "long int" कीवर्ड का उपयोग करके वैरिएबल बना सकते है | इस कीवर्ड की मदद से बना वेरिएबल नार्मल वेरिएबल से ज्यादा मात्रा में डेटा स्टोर कर सकता है |

Example

```c
#include <stdio.h>

int main()
{
 int x =5;
 long int y;
 printf("Enter a Number\n");
 scanf("%d",&y);
 printf("value of x = %d , y = %d ",x,y);

 return 0;
}
```

Character Type

- Character Type से बने वेरिएबल में हम कैरेक्टर टाइप के डेटा को स्टोर करते है |
- "char" कीवर्ड का उपयोग कैरेक्टर टाइप के वेरिएबल को बनाने करने के लिए क्या जाता है |
- "char" कीवर्ड से बना वेरिएबल एक सिंगल करैक्टर को स्टोर करता है |
- Character Type से बने वैरिएबल की साइज 1 byte होती है |
- char डेटा टाइप से बने वेरिएबल की वैल्यू को प्रिंट करने के लिए %c Format Specifier का उपयोग किया जाता है |

Example

```c
#include <stdio.h>
int main()
 {

 char ch = 'A';
 char ch2;
 printf("Enter a Character \n");
 scanf("%c",&ch2);
 printf("value of ch = %c , ch2 = %c ",ch,ch2);

return 0;
}
```

Floating Point Data Type

Floating point data type दो प्रकर के होते है -:

- Float
- Double

Float

- यह डेसीमल पॉइंट वाले डेटा को स्टोर करने के लिए उपयोग किया जाता है |
- "float" कीवर्ड का उपयोग floating point type के वेरिएबल को बनाने करने के लिए किया जाता है |
- Float Type से बने वैरिएबल की साइज 4 byte होती है |
- Float डेटा टाइप से बने वेरिएबल की वैल्यू को प्रिंट करने के लिए %f Format Specifier का उपयोग किया जाता है |

Example of Float Data Type

```c
#include <stdio.h>
int main()
{
 float x =5.50;
 float y;
 printf("Enter a decimal Number\n");
 scanf("%f",&y);
 printf("sum of x and y is %f ",x+y);

return 0;
}
```

Double

- यह भी फ्लोट डेटा टाइप की तरह डेसीमल पॉइंट वाले डेटा को स्टोर करने के लिए उपयोग किया जाता है |
- "double" कीवर्ड का उपयोग double data type के वेरिएबल को बनाने करने के लिए किया जाता है |
- double Type से बने वैरिएबल की साइज 8 byte होती है |
- double डेटा टाइप से बने वेरिएबल की वैल्यू को प्रिंट करने के लिए **%lf** Format Specifier का उपयोग किया जाता है |

Example of Double Data Type

```c
#include <stdio.h>
int main()
{
 float  x = 679999999.454;
 double y = 679999999.454;

 printf("float x = %f and double y = %lf ",x,y);

 return 0;
```

2. Derived Data Types

Derived Data Types ऐसे Data Types होते है जिनमे वेरिएबल की ग्रुपिंग होती है | Derived Data Types निम्नलिखित है -:

- Array
- Function
- Pointer

3. User-Defined Data Types

ऐसे Data Types जो पहले से डिफाइंड नहीं होते तथा जिनको अलग से डिफाइंड करना पड़ता है **User-Defined Data Types** कहलाते है | जैसे कि – Structure, Enumerator, union |

सी लैंग्वेज में कंट्रोल स्टेटमेंट क्या है? (What is Control Statements In C In Hindi)

ऐसे **statements** जिससे हम किसी प्रोग्राम के **flow** को निर्धारित करते है, **Control Statements** या **Decision control statements** कहलाते है |

किसी भी प्रोग्रामिंग लैंग्वेज में, **condition** के आधार पर विभिन्न कार्यों को करने की आवश्यकता होती है और ऐसे कार्यों को करने के लिए हम **Control Statements** या Decision making statements का उपयोग करते है |

उदाहरण के लिए, एक ऑनलाइन वेबसाइट पर विचार करें, जब आप गलत आईडी या पासवर्ड दर्ज करते हैं तो यह error page प्रदर्शित करता है और जब आप सही क्रेडेंशियल दर्ज करते हैं तो यह welcome page प्रदर्शित करता है।

तो वहाँ एक logic होना चाहिए जो condition (आईडी और पासवर्ड) की जांच करता है और यदि condition सही है तो यह welcome page प्रदर्शित करता है अन्यथा यह error page प्रदर्शित करता है |

कंट्रोल स्टेटमेंट के प्रकार (Types of Control Statement in C In Hindi)

C language में हमारे पास निम्नलिखित decision control statements है

1. If statements
2. Loop Statements
3. Switch Statement
4. Conditional Operator Statement
5. Goto Statement

If statements

If statement, सी लैंग्वेज में एक शक्तिशाली conditional statement में से एक है। If statement का उपयोग हमेशा किसी condition के साथ किया जाता है।

If statement की बॉडी के अंदर किसी भी statement को execute करने से पहले condition को check किया जाता है।

Syntax

```
if (condition)
{
  Statements;
}
```

if else Statement

इस decision control statement में, हमारे पास statements के दो block होते हैं। एक ब्लॉक if के अंदर होता है और दूसरा ब्लॉक esle के अंदर होता है |

यदि If condition सही (True) होता है, तो if block के अंदर का स्टेटमेंट execute हो जाता है, नहीं तो else वाले ब्लॉक के अंदर का स्टेटमेंट execute होता है |

Syntax

```c
if (Condition)
{
    True block of statements
}
Else
{
    False block of statements
}
```

Loop Statements

जब हम किसी खास कंडीशन के पूरा होने तक किसी पर्टिकुलर स्टेटमेंट को बार बार चलाना चाहते है तब उस स्थिति में हम **Loop Statements** का उपयोग करते है |

सी लैंग्वेज में लूप स्टेटमेंट तीन तरह के होते है :

- While Loop
- Do While Loop
- For Loop

While loop

While loop में सबसे पहले कंडीशन चेक होता है और यदि कंडीशन सही होता है तभी while loop के अंदर का स्टेटमेंट रन होता है while loop के अंदर का स्टेटमेंट तब तक चलता है जब तक while loop का कंडीशन True होता है जैसे ही कंडीशन False हुवा कंट्रोल while loop से हट जाता है और कोई दूसरा स्टेटमेंट Execute होता है |

Do while loop

do while loop भी while loop की तरह ही होता है मगर इसमें एक अंतर ये होता है की while loop में पहले कंडीशन चेक होता है और कंडीशन True होने पर ही while loop के अंदर का Statements रन होता है जबकि do while loop में कंडीशन True हो या False उसके अंदर का स्टेटमेंट एक बार तो चलता ही है |

Syntax

```c
Do
{
//statements inside the loop
}
While(condition);
```

For loop

For Loop में हम parentheses " () " के अंदर वेरिएबल का initialization और कंट्रोल कंडीशन दोनों एक साथ लिखते है | यदि For Loop का कंडीशन सही होता है तो उसके अंदर का statement रन होता है नहीं तो स्टेटमेंट रन नहीं होता |

Syntax

```c
for( initialization statement; condition)
{
//statements inside the loop
}
```

Switch Statement

जब हमारे पास काफी सारे कंडीशन होते है और हमें उस कंडीशन से मैच कर रहे स्टेटमेंट को रन करना होता है तब हम स्विच केस स्टेटमेंट का उपयोग करते है | if else के द्वारा किया जाने वाला सभी काम हम **Switch Case Statement** की मदद से कर सकते है |

Syntax

```c
Switch(expression)
{
Case label1:
            Statement(S);
            Break;
Case label2:
            Statement(S);
            Break;
Case label3;
            Statement(s);
            Break;
….
Case labelN:
           Statement(s);
```

Goto Statement

goto statement को jumping control statement के नाम से भी जाना जाता है इसका उपयोग प्रोग्राम के कंट्रोल को एक ब्लॉक से दूसरे ब्लॉक में ट्रांसफर करने के लिए किया जाता है | goto statement को डिक्लेयर करते समय goto कीवर्ड का उपयोग किया जाता है |

Syntax

```
goto  labelname;
Labelname;
```

उपरोक्त syntax में, goto एक कीवर्ड है जिसका उपयोग labelname पर control स्थानांतरित करने के लिए किया जाता है। labelname एक variable name है। goto प्रोग्राम के control को labelname में स्थानांतरित कर देगा और labelname के बाद वाले statements को execute किया जाएगा।