

## # History of C programming Language :-

- ① The first most high level language was ALGOL. ALGOL was introduced in 1950. It is also called the base or the father of programming languages. ALGOL introduced the concept of Structured programming. It mainly used the concept of Sequence, Decision and Repetition.
- ② Now using this ALGOL a new programming language is introduced name as CPL. Stand for Common Programming Language in 1967. It is developed by Christopher Strachey.
- ③ Using ALGOL and CPL. The Martin Richards introduced BCPL Stand for Basic Combined programming language in 1967. It is very popular and it was used for designing and developing System software.
- ④ Now using this three programming language A new programming language is introduced which is called "B". B programming language was developed by Ken Thompson at AT&T Bell Laboratory in 1970. Here Dennis Ritchie and Ken Thompson was developing UNIX operating system. Initially they use PDP 7 chip set but after moving on PDP 11 chip set. This programming language does not use the powerful features of this chip set. Now they thought we have to improve this language.

⑤ Now Dennis Ritchie Come here and improved this programming language "B" To introduce new programming language "C" in 1972 C programming language has all the powerful features which is not available in previous languages C programming language is very popular at that time and it is also used to write UNIX operating system

⑥ Now Brian Kernighan and Dennis Ritchie wrote a book together on C program in 1978

⑦ The Commercial version of C was introduced by ANSI/ISO in 1989-90  
ANSI → American National Standard Institute

ALGOL 1950 → CPL 1967 → BCPL 1967 → B 1972

→ C 1972 → KSR C 1978 → ANSI/ISO 1990

→ C99 [1999] → C11 [2011] → C18 [2019]

## # Introduction of C :-

C is a middle level, procedural oriented programming language developed by "Dennis Ritchie" at AT&T bell lab in the year 1972 in USA.

Syntax :-  
Preprocessor Statement  
Program #include < stdio.h > → Standard Input output library.  
Void main () {  
    // code  
}

Note :- 1) C is a Case-sensitive programming language.

2) C is an extension of C language files.

3) C language support 32 Keywords.

## Advantage :-

- ①. Open sources.
- ②. Easy to learn.
- ③. Portable.
- ④. Easy debugging.
- ⑤. Dynamic memory allocation.

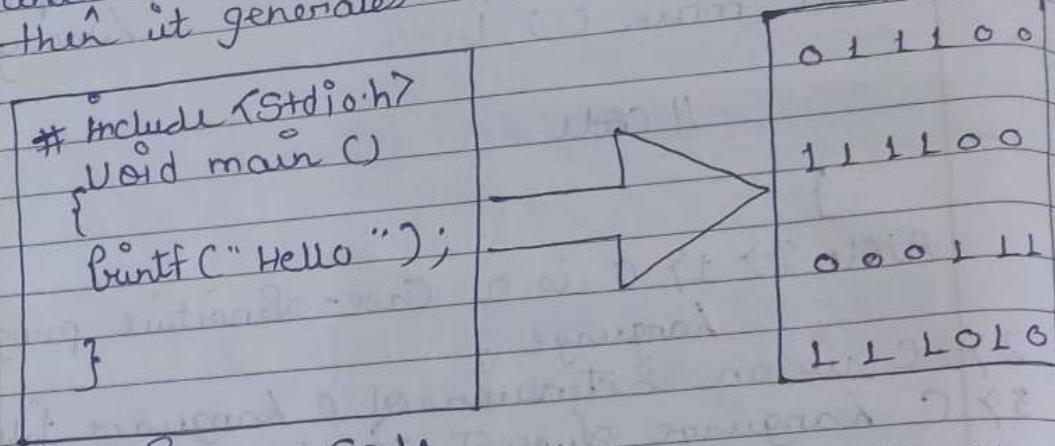
## Dis-Advantage :-

- ①. Object oriented.
- ②. Exception Handling.
- ③. Garbage Collector.
- ④. Platform dependent.

## # Compilation process in C :-

The compilation is a process of converting the source code into object code. It is done with the help of the compiler. The compiler checks the source code for the syntactical or structural errors, and if the source code is error-free, then it generates the object code.

- 1
- 2
- 3
- 4



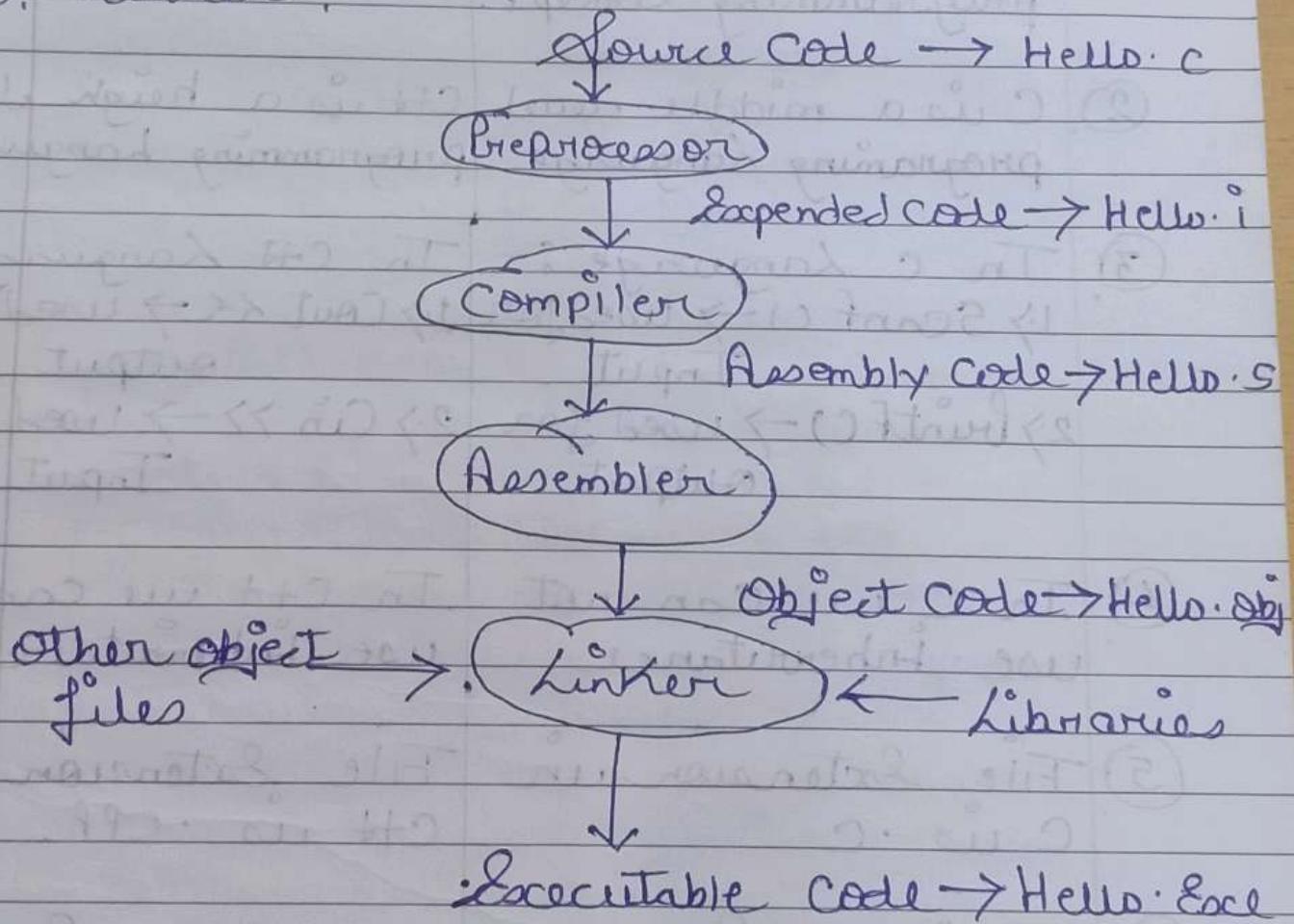
### Source Code

The C compilation process converts the source code taken as input into the object code or machine code. The compilation process can be divided into four steps. For example → Pre-processing, Compiling, Assembling and linking.

The pre-processor takes the source code as an input, and it removes all the comments from the source code. The pre-processor takes the preprocessor directive and interprets it.

The following are the phase through which our program passes before being transformed into an executable forms.

- (1) Preprocessor
- (2) Compiler
- (3) Assembler
- (4) Linker



## # Difference between 'C' and 'C++' :-

C	C++
①. C supports function programming or procedural oriented programming Concept.	C++ supports object oriented programming Concept.
②. C is a middle level programming Language	C++ is a high level programming languages.
③. In C Language :- 1> scanf() → used for Input. 2> printf() → used for output.	In C++ Languages :- 1> cout << → used for output. 2> cin >> → used for Input.
④. In C we can not use inheritance	In C++ we can use inheritance
⑤. File Extension in C is .C	File Extension in C++ is .CPP.
⑥. C is a subset of C++.	C++ is a superset of C.

# Data-type in C  $\Rightarrow$  Data-type is a type of data which is used in the program. In other words we can say that it is used to declare a variable.

OR

Data-type specify how we entered data in our program and what type of data we entered.

There are two type of Data-types:-

### Data-types

#### Primary Data-type

	Size	Range
- int	2 byte	-32768 to 32767
- char	1 byte	-128 to 127
- float	4 byte	$3.4 \times 10^{-38}$ to $3.4 \times 10^{38}$
- double	8 byte	
- void	no size	

#### Secondary Data-type

- Array
- Pointer
- Structure
- Union

# Variable in C :- The name of memory location is called variable. It is used to store data. Variable are changeable. We can change the value of variable during the execution of program and variable can be reused many times.

Rules for defining variable :-

- ① A variable can have alphabets, digits and underscore.
- ② A variable name can start with the alphabet, and underscore only. It can not start with a digit.
- ③ No whitespace is allowed within the variable name.
- ④ A variable name must not be any Reserved Word or Keyword. For example int, float etc.

Syntax :-

Data Type Variable-name ;

Example :-

int a;	→	%d
float b;	→	%f
char c;	→	%c
Double d;	→	%d.

# printf() and scanf() in C :-

The printf() and scanf() functions are used for input and output in C language. Both functions are built-in library functions, defined in stdio.h [header file].

printf() Function :- The printf() function is used for output. It prints the given statement to the console. The syntax of printf() function is given below.

Syntax :-

```
printf ("Hello World");
```

The format string can be %d [integer], %c [character], %s [string], %f [float] etc.

Example :- printf ("%d", q);

Output → 10

scanf() Function :- The scanf() function is used for input. It reads the input data from the console.

Syntax :- scanf ("%d", &q);

Example :- int q;

```
printf ("Enter value of q");
scanf ("%d", &q);
```

Output :- Enter value of q  
→ User Input.

## # Keyword in C Language →

A Keyword is a reserved word. You can not use it as a variable and constant name etc. There are only 32 reserved words [Keyword] in the C language.

OR

Keyword is a reserved words whose meaning is already define in the Compiler.

Example → int.

A list of 32 Keyword in the c language is given below:-

①. auto	Case	Const	Default
②. double	Enum	Float	Double
③. int	Register	Short	Signed
④. struct	TypeDef	Unsigned	Volatile
⑤. break	Char	Continue	Do
⑥. Else	Extern	For	If
⑦. long	Return	Signed	Static
⑧. Switch	Union	Void	While

## # Identifier in C Language →

A Identifier refers to name that is used to identify Variable, function, Structure and so on.

Example → int sal;

double money ; } Variable

Function → { void fun ()  
; = }

## Rules for Constructing C Identifiers:

- ①. The first character of an identifier should be either an alphabet or an underscore, and then it can be followed by any of the character digit or underscore.
- ②. It should not begin with any numerical digit.
- ③. In identifier, both uppercase and lowercase letter are distinct. Therefore we can say that identifier are case Sensitive.
- ④. Commas or blank space cannot be specified within an identifier.
- ⑤. Keywords can not be represented as an identifier.
- ⑥. The length of the identifier should not be more than 31 Character.

## # Difference between Keyword and identifier:

Keyword	Identifier
①. Keyword is a predefined word	Identifier is a user defined words.
②. It must be written in a lower case letter	It can be written in both lowercase and uppercase letters.
③. Its meaning is predefined in the C Compiler.	Its meaning is not defined in the C Compiler.

④ It is a combination of alphabetical characters.

It is a combination of alphanumeric characters.

⑤ It does not contain the underscore characters.

It can contain the underscore character.

# Tokens in C Language →  
Tokens are the smallest unit of a program we can define the token as a smallest individual elements.

Classification of Tokens :-

① Keyword → 32 Key in C

② Identifier → int a.

③ String → "Hello"

④ Operator → +, -, \*, %, <, >, =, ==

⑤ Constant → a=5 [fixed values]

⑥ Special Symbol → ; , : { } [ ]

Example → #include <stdio.h>

Void main()

int a, b, c;

printf("Enter value of a");

scanf("%d", &a);

printf("Enter value of b");

scanf("%d", &b);

c = a+b;

printf("%d", c);

Operators in C :- An operator is simply a symbol that is used to perform operation. There can be many types of operation like arithmetic and bitwise etc.

There are following types of operator to perform different types of operation in C language.

- ① Arithmetic Operators
- ② Relational Operators.
- ③ Logical Operators.
- ④ Increment & Decrement Operators.
- ⑤ Assignment Operators.
- ⑥ Bitwise Operators.

Symbols For Operator →

① Arithmetic Operators → The arithmetic operator supported by the C language. Assume variable A holds 10 and variable B holds 20 then

Operators → +, -, \*, /, %

Example → Here  $A = 10$  and  $B = 20$

$$\text{then } [A+B] \rightarrow 10+20 \rightarrow 30$$

$$[A-B] \rightarrow 10-20 \rightarrow -10$$

$$[A*B] \rightarrow 20 * 20 \rightarrow 200$$

$$[A/B] \rightarrow 20 / 10 \rightarrow 2$$

$$[B \% A] \rightarrow 20 \% 10 \rightarrow 0$$

② Relational Operators  $\Rightarrow$  The relational operator supported by C language. Assume variable A holds 10 and variable B holds 20 then

Operator  $\rightarrow$   $<$ ,  $>$ ,  $=$ ,  $\geq$ ,  $\leq$ ,  $\neq$

Example  $\rightarrow$  Here  $A = 10$  and  $B = 20$   
 then  $[A < B] \rightarrow [10 < 20] \rightarrow \text{True} \rightarrow 1$   
 $[A > B] \rightarrow [10 > 20] \rightarrow \text{False} \rightarrow 0$

③ Logical Operators  $\Rightarrow$  The logical operators supported by C language. Assume variable A holds 1 and variable B holds 0 then

Operator  $\rightarrow$   $\&&$ ,  $\|$ ,  $!$   
 AND , OR , NOT

Example  $\rightarrow$  Here  $A = 1$  and  $B = 0$   
 then  $[A \&& B] \rightarrow [1 \&& 0] \rightarrow \text{False} \rightarrow 0$   
 $[A \| B] \rightarrow [1 \| 0] \rightarrow \text{True} \rightarrow 1$   
 $! [A] \rightarrow ! [1] \rightarrow \text{False} \rightarrow 0$

④ Increment & Decrement Operator  $\Rightarrow$

There are two type  
 of Increment and Decrement Operators

Increment  $\rightarrow$  Pre Increment  
 $\rightarrow$  Post Increment

Decrement  $\rightarrow$  Pre Decrement  
 $\rightarrow$  Post Decrement

Operators  $\rightarrow$  Pre Incr  $\rightarrow$   $++A$   
 Post Incr  $\rightarrow$   $A++$   
 Pre Decre  $\rightarrow$   $--B$   
 Post Decre  $\rightarrow$   $B--$

- i) Pre Incr  $\rightarrow$   $++A$   $\rightarrow$  first Increment then print
- ii) Post Incr  $\rightarrow$   $A++$   $\rightarrow$  first print then Increment
- iii) Pre Decre  $\rightarrow$   $--B$   $\rightarrow$  first Decrement then print
- iv) Post Decre  $\rightarrow$   $B--$   $\rightarrow$  first print then Decrement.

Example  $\rightarrow$  Here  $A = 10$  and  $B = 20$   
 then  $[A++] \rightarrow 11$   
 $[B--) \rightarrow 19$

⑤ Assignment Operator  $\rightarrow$  The assignment operator supports by the C language. There are two types of assignment Operator are as follow.

i) Simple assignment Operator  $\rightarrow [=]$

Example  $\rightarrow$  Here  $A = 10 \rightarrow$  Right to Left

ii) Compound assignment Operators  $\rightarrow$   
 $+ =$ ,  $- =$ ,  $* =$ ,  $/ =$

Example  $\rightarrow$  Here  $A = 10$  then  
 $A += 10 \rightarrow A = 20$   
 $A -= 5 \rightarrow A = 5$

⑥ Bitwise Operator  $\Rightarrow$  Bitwise Operator  
works on bits and  
perform bit-by-bit operators

Operator  $\rightarrow$  &, |, !, ^, ~  
AND, OR, NOT, XOR, One Comp

# Control statements in C  $\Rightarrow$   
There are two types of  
Control statement in C are as follow

- ① Conditional Statement.
- ② Decision making Statement.

Conditional Statement :- There are five types of Conditional Statement are as follow.

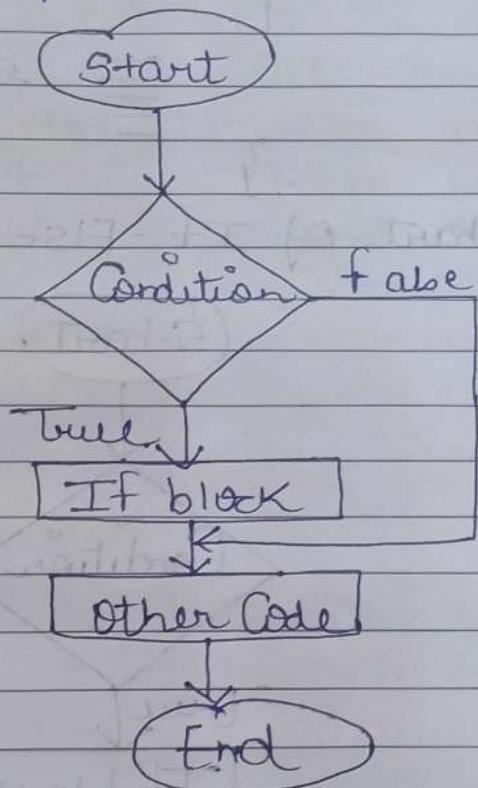
- ① If Statement.
- ② If-Else Statement.
- ③ If-Else-If Ladder Statements.
- ④ Nested If-Else Statements.

If Statement  $\Rightarrow$  It is used when we want to test a condition given Condition is true If block will be executed otherwise skip remaining code. It is also used when there is only one condition and one statement.

Syntax  $\Rightarrow$  If (Condition / Expression)

= Statement or Code to be Executed.

Flowchart of If statement in C  $\Rightarrow$



If - Else Statements  $\Rightarrow$  It is used to perform two

operations for a single condition. If the given condition is true then if block will execute otherwise else block execute.

It is also used when there is only one condition and two statements.

Syntax :> If ( condition / expression )  
                  ≡ Statement ①.

}

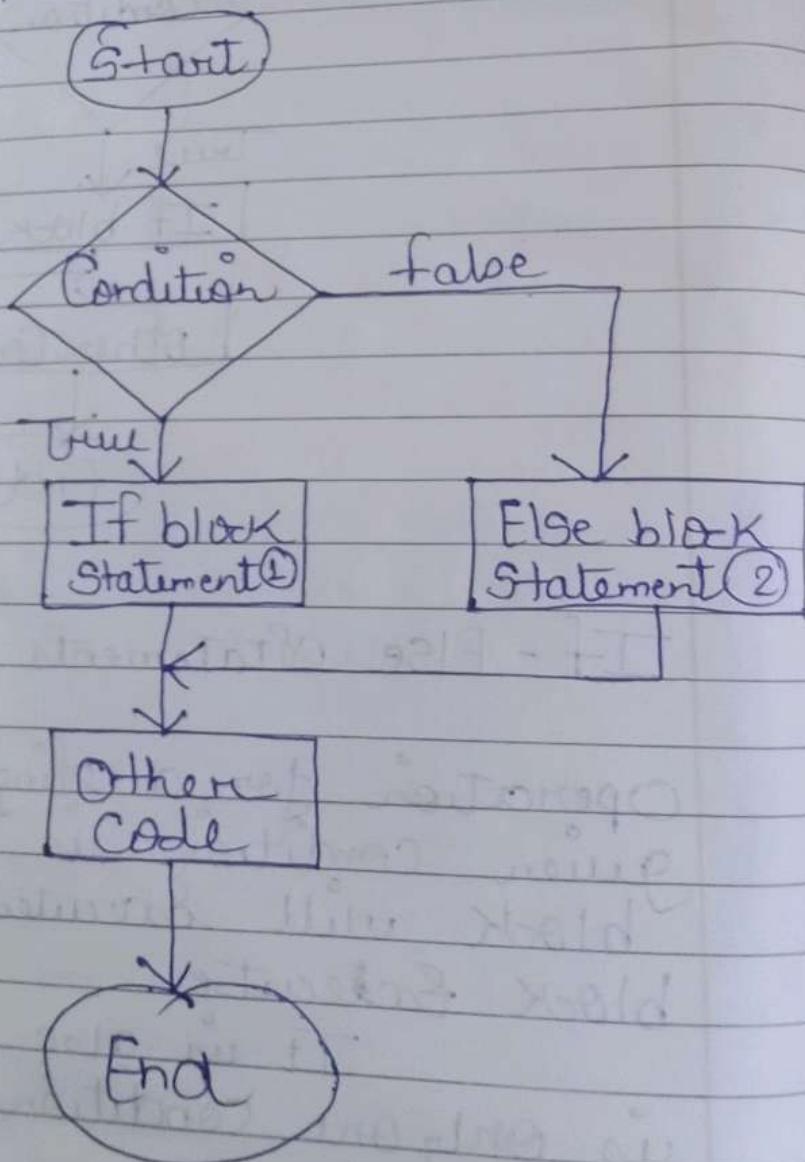
Else

{

                  ≡ Statement ②.

}

Flowchart of If - Else Statement in C :



If Else-If Ladder Statement  $\Rightarrow$  The If Else-If ladder statement is an extension to the If-Else statement. It is used in the scenario where there are multiple cases to be performed for different condition.

In If Else-If ladder statement If a condition is true then the statement defined in the If block will be executed. Otherwise If some other condition is true then the statement defined in the Else-If block will be executed, at the last If none of the condition is true then the statement defined in the Else block will be executed. There are multiple Else-If blocks possible.

Syntax  $\Rightarrow$  If { condition ① }

    ≡ Statement ①

    } Else-If { condition ② }

        ≡ Statement ②

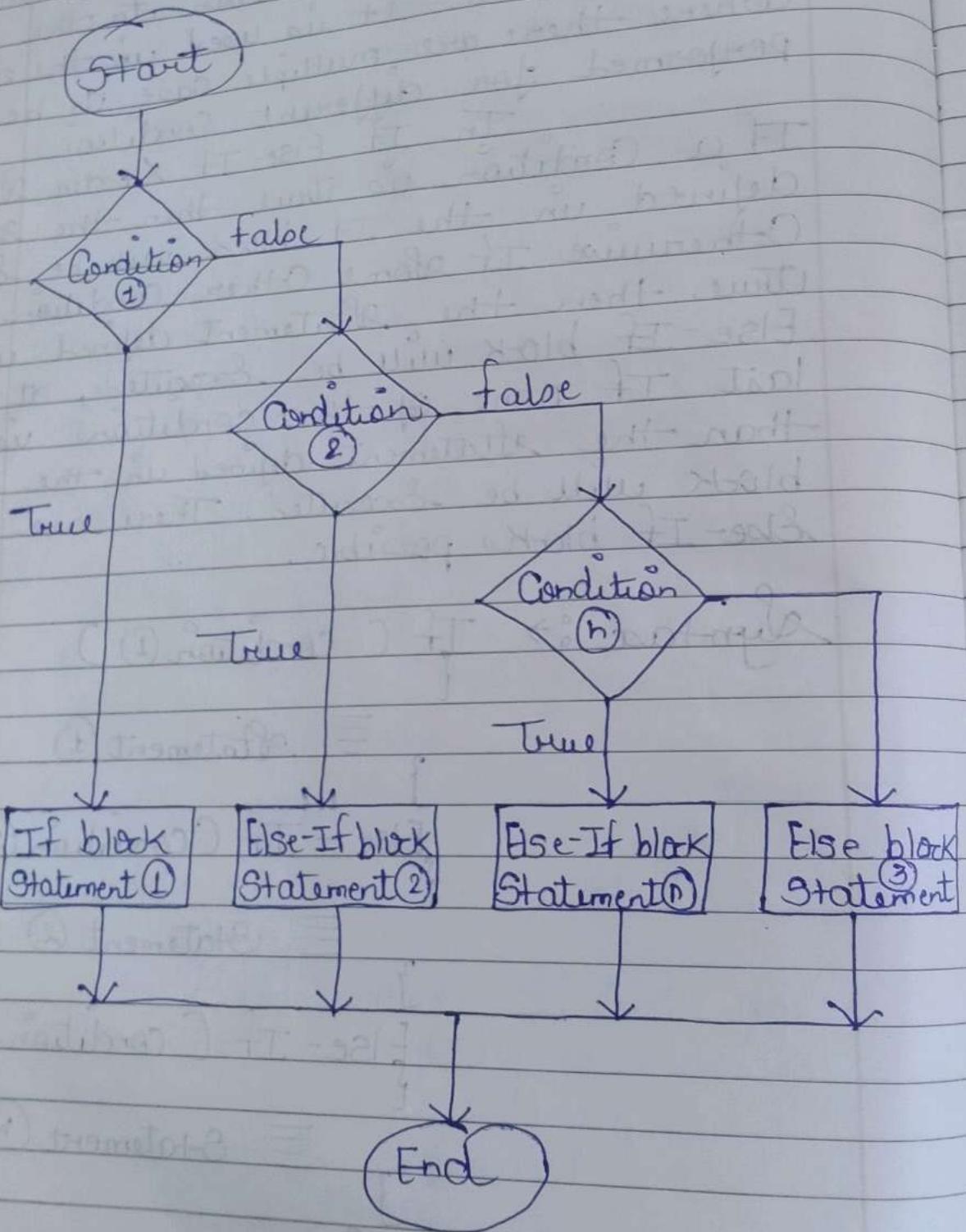
    } Else-If { condition ③ }

        ≡ Statement ③

    } Else

        ≡ Statement ④

Flowchart of Else-If Ladder Statement in C ↳



Nested If-Else Statement  $\Rightarrow$  whenever If Statement Contains itself another If and Else Statement Or Else Statement Contains itself ~~other~~ another If and Else Statement than it is Known as nested If and nested Else Statements

Syntax  $\Rightarrow$  If ( Condition )

{ If ( condition ) }

= Statement of nested If

{ Else

} = Statement of nested Else

{ Else

{ If ( condition ) ,

= Statement of nested

If

{ Else

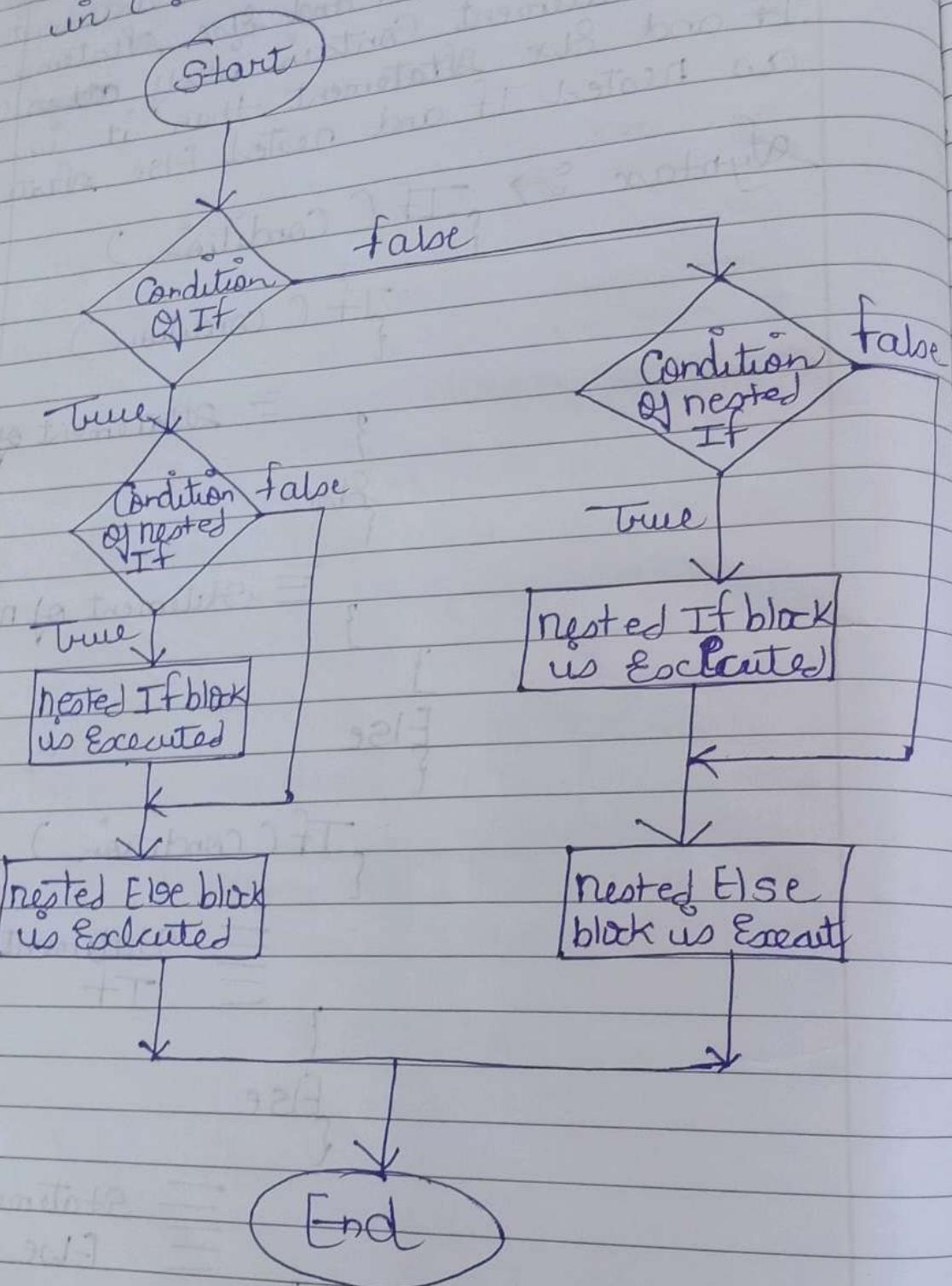
= Statement of nested

Else

}

}

# Flowchart of Nested If-Else Statements in C



# Decision Making Statements  $\Rightarrow$  There are one type of decision making statement are as follow:-

① Switch Statement  $\Rightarrow$  The Switch Statement in C is an alternate to If-Else-If Ladder Statement which allows us to execute multiple operations for the different possible values of a single Variable called Switch Statements. Here we can define various statements in the multiple cases for the different values of a single variable.

The Syntax of Switch Statement in C language is given below:-

Syntax :- { Switch (Condition / Expression)

case 1 :

{     ≡ Statements ; break ;  
    }

Case 2 :

{     ≡ Statements ; break ;  
    }

Case n :

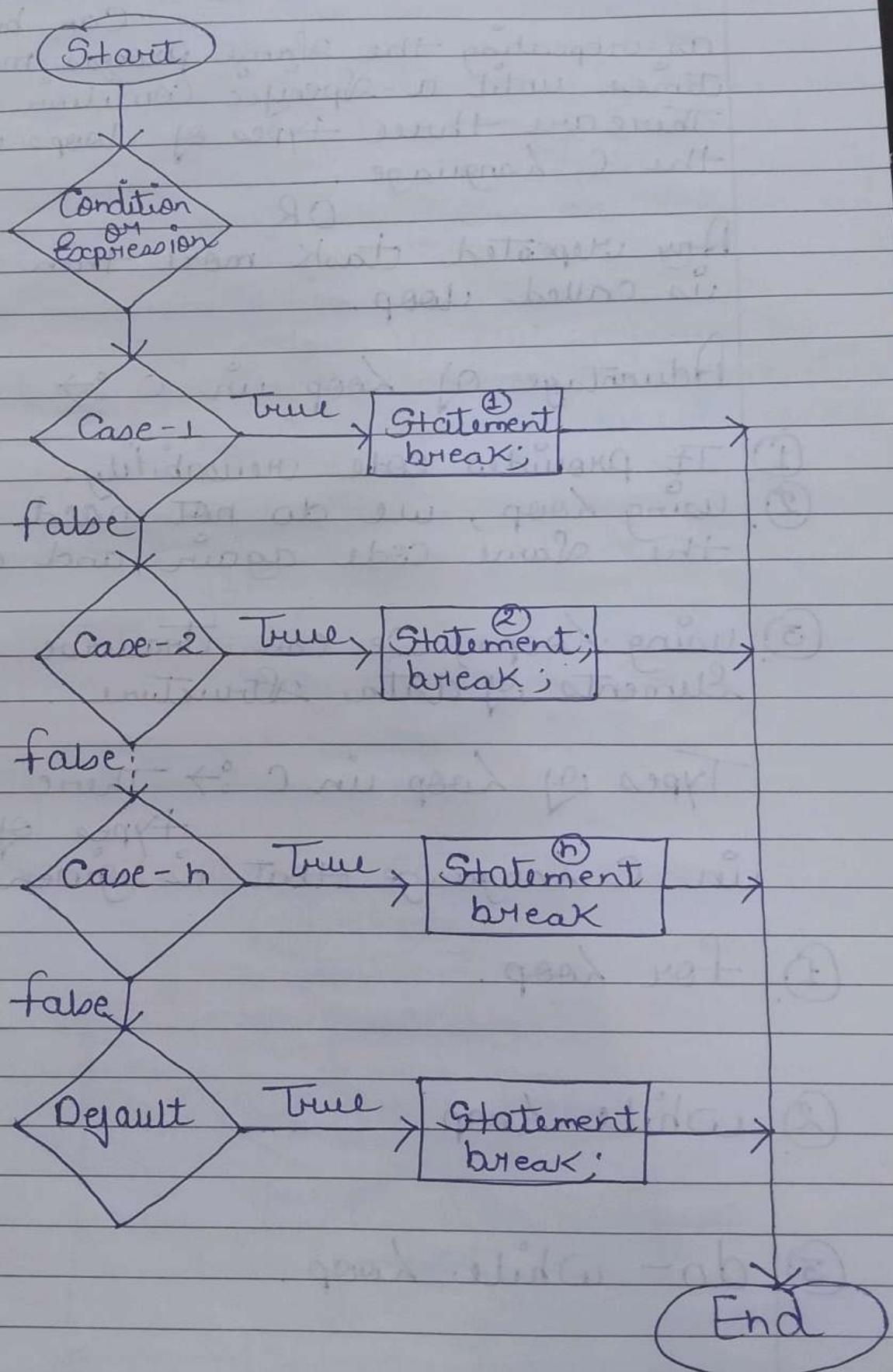
in last  $\rightarrow$  default.  
{     ≡ Statements  
    }

## Rules for switch statements in C language

- ① The switch expression must be of an integer or character types.
- ② The case value must be an integer or character constant.
- ③ The case value can be used only inside the switch statements.
- ④ The break statement in switch case is not must. It is optional. If there is no break statement found in the case, all the cases will be executed present after the matched case. It is known as fall-through the state of C switch statements.

In programming :→

## Flow chart diagram of switch statements



# Loop in C language :> The looping can be defined as repeating the same process multiple times until a specific condition satisfies. There are three types of loops used in the C language.

(1)

OR  
Any repeated task more than one time is called loop.

Advantage of Loop in C :>

- ① It provides code reusability.
- ② Using loop, we do not need to write the same code again and again.
- ③ Using loops, we can traverse over the elements of data structure.

Types of Loop in C :> There are three types of loops in C language that is given below.

- ① for loop.
- ② while loop.
- ③ do-while loop.

① For loop :- The for loop is used in the case where we need to execute some part of the code until the given condition is satisfied. The for loop is also called as a pre-tested loop. It is better to use for loop if the number of iteration is known in advance.

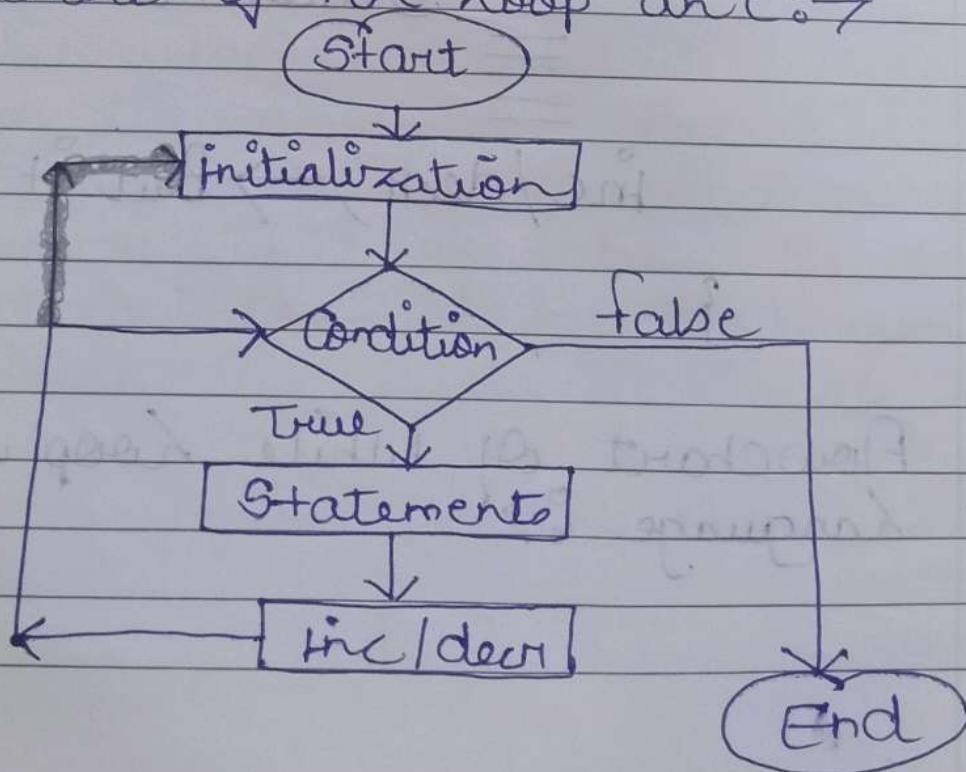
OR

A for loop enable us to perform n numbers of steps together in one line. In for loop, a loop variable is used to control the loop.

Syntax :-

```
for (Initialization ; Condition ; Inc/Dec)
{
    Statements
}
```

Flowchart of For loop in c :-



② While loop in C  $\Rightarrow$  while loop is also known as a pre-tested loop. In general, a while loop allows a part of the code to be executed multiple times depends upon a given condition. It can be viewed as a repeating If statements. The while loop is ~~used~~ mostly used in the case where the number of iteration is not known in advance.

OR:

while loop is a pre-test loop, it first test a specified conditional expression and as long as the conditional expression is true then loop body statements will be executed.

Syntax  $\Rightarrow$

while (Conditional Expression)

{

====

Statements

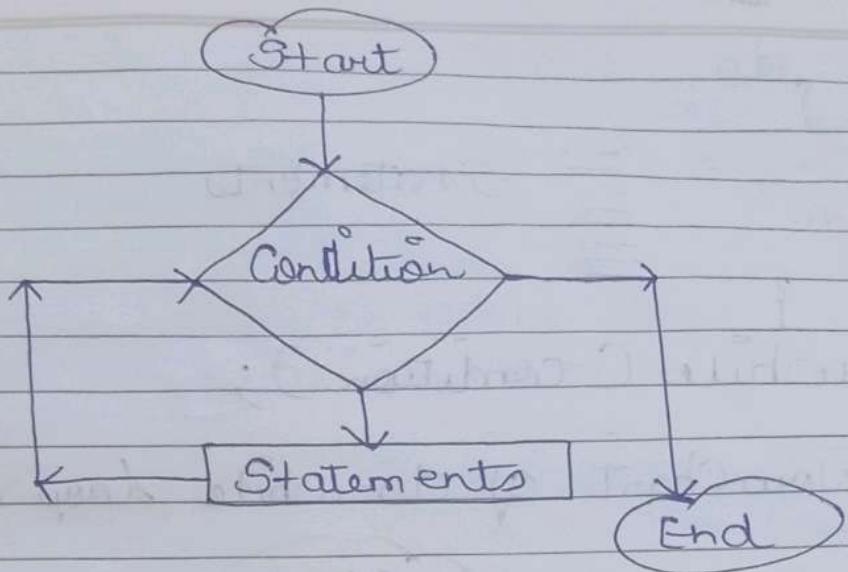
====

====

inc/dec;  $\rightarrow$  But it is not must

}

Flowchart of while loop in C  
language  $\Rightarrow$



③. Do - while loop in C :→ The do-while loop is a post tested loop. Using the do-while loop, we can repeat the execution of several parts of the statements. The do-while loop is mainly used in the case where we need to execute the loop at least once. The do-while loop is mostly used in menu driven programs. Where the termination condition depends upon the end user.

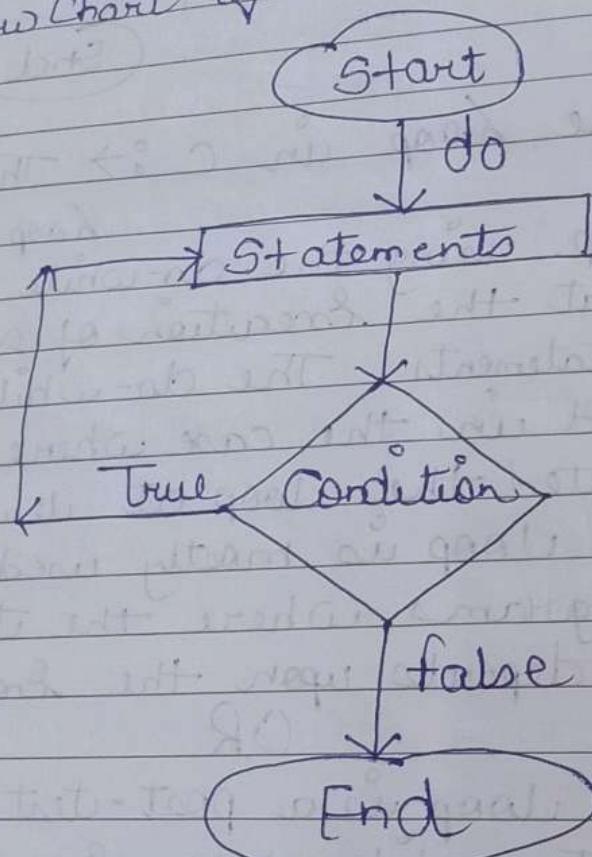
OR

Do-while loop is a post-test loop. It is similar to while loop except it executes its body at least once whether the condition is true or false. The do-while loop terminates when the test expression is evaluated to be zero.

Syntax of do-while loop in C language :→

{ do  
  ≡ statements  
  {  
    while C condition);

Flowchart of do-while loop in C →



# Function in C language  $\Rightarrow$  In C, we can divide a large program into the basic building block function is the block of statements enclosed by {}, which can call by another function is called function. A function can be called multiple times to provide reusability and modularity to the C program.

General Definition  $\Rightarrow$  A function is a set of statements that takes inputs, do some specific computation and produce output is known as function.

function Aspects  $\Rightarrow$  There are three aspects of a C function.

①. function declaration  $\Rightarrow$  A function must be declared globally in a C program to tell the compiler about the function function name, function parameter and return type.

②. function call  $\Rightarrow$  function can be called from ~~anywhere~~ main function only.

③. function definition  $\Rightarrow$  It contains the actual statement which are to be executed. It is the most important aspect to which the control comes when the function is called. Here, we must notice that only one value can be returned from the function.

**Syntax**  $\Rightarrow$  ① Function Declaration  $\Rightarrow$  argument list  
Return type function name ( );  
② Function Definition  $\Rightarrow$  Return type function name ()  
function Body

③ Function Calling  $\Rightarrow$  function name ( );

# Types of Function  $\Rightarrow$  There are two types of function in C programming are as follows:

① Pre-defined or Library Function  $\Rightarrow$  Pre-defined functions are the functions which are declared in the C header file such as scanf(), printf() etc.

② User-defined Function  $\Rightarrow$  User-defined functions are the functions which are created by the C programmer, so that he can use it many times. It reduces the complexity of a big program and optimizes the code.

# Different Aspects of Function calling  $\Rightarrow$   
A function may or may not accept any arguments. It may or may not return any values. Based on these facts, there are four

different aspects of function calls.

- (1) function without arguments and without Return values.
- (2) function without arguments and with Return values.
- (3) function with arguments and without Return values.
- (4) function with arguments and with Return values.

## # Introduction of C++ programming language →

C++ is a high-level, general purpose object oriented programming language developed by "Bjarne Stroustrup" at Bell Lab of AT&T in the year of 1980 in USA.

Syntax :-

```
#include <iostream.h> → Header declaration  
Using namespace std;  
int main () → Control declaration  
{  
    // Code  
}
```

Note :-

- 1) C++ is a case-sensitive language.
- 2) .CPP is an extension of C++ files.
- 3) C++ supports 32+ keywords.
- 4) C++ is an superset of C language.

## # Features of C++ programming language →

- 1) Easy to learn.
- 2) Open Source.
- 3) portable → OS System Configuration Same.
- 4) OOP's Concept:
  - Class & Object.
  - Inheritance.
  - Polymorphism.
  - Encapsulation.
  - Abstraction.

- 5) Memory management.
- 6) Rich library.
- 7) Exception Handling.

## # Difference between programming & Coding :

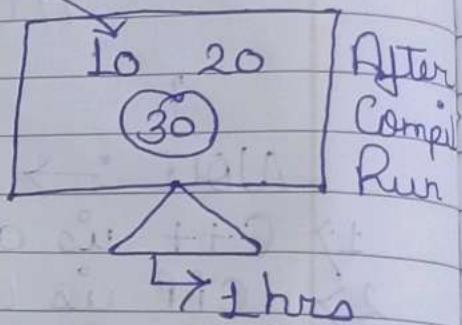
① In programming we think how to solve the problem.

## Coding

In Coding: We think about how to implement the problem in computer.

```
main
{
    int a, b, c;
    c = a+b;
    cout << c;
}
```

→ 10 hrs



## # Difference between Statically typed language & Dynamically typed language :

**Statically**

1) The language which force the user to define the datatype before initialization of a variable is called **Statically typed language**.

Example → C, C++, Java  
`int a=10;  
Print(a);`

**Dynamically**

The language in which the datatype is not compulsory and its optional is called **dynamically typed languages**.

Example → Python, JavaScript

## # Data Type in C++ Programming Language →

Datatype define the type of value means which kind of value the variable will store.

Example : > " int " a - 10 ;  
datatype

### Types of Datatype in C++ →

There are three types of Datatypes in C++ programming language are as follow :-

- 1> Pre-defined → integer, boolean, float, double etc
- 2> Derived → Function, Array, pointer etc.
- 3> User-defined → class, structure, union etc.

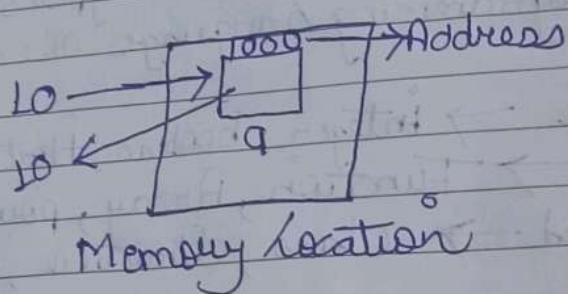
#### Pre-defined Data types →

Data Type	size	Range
1> int	4 bytes	-2147483648 to 2147483647
2> Char	1 byte	0 to 255
3> boolean	1 byte	True or false
4> float	4 bytes	Storing 6 to 7 decimal digits
5> double	8 bytes	Storing 15 decimal digits
6> String	8 bytes	" --- " ;

# Variable in C++ Programming Languages ↗  
Variable is nothing but name of  
memory location where we store the values

Syntax ↗ Data-type Variable\_name ;  
Data-type Variable\_name = value ;

Example ↗ int a = 10 ;  
char b = 'B' ;



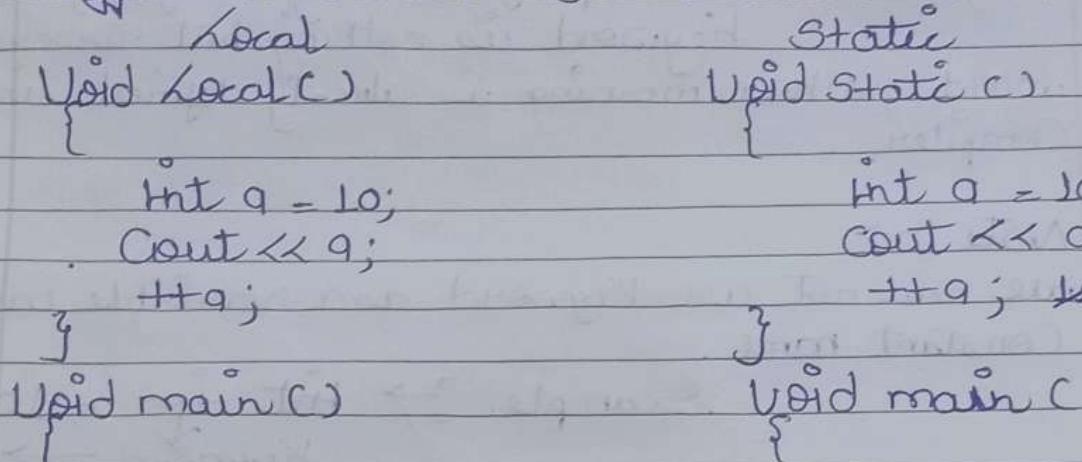
Note ↗

- 1) Variable are the Case-Sensitive in C++ .
- 2) In C++, Variable must be start with either (a-z , A-Z) or underscore (-) .
- 3) we can not give extra space between the variable .

Types of Variable in C++ ↗ There  
are three types of Variable  
in C++ programming language are as  
follow ↗

- ① Global Variable .
- ② Local Variable .
- ③ Static Variable .

Difference between static or local variable:-



`local(); → 10`  
`local(); → 10`  
`local(); → 10`

`Static() → 10`  
`Static() → 11`  
`Static() → 12`

1> Local variable are declared inside the function.

2> The default value of local variable is garbage [Random Integer number].

3> life time of local variable are still the block is active.

1> Static variable are declared inside or outside of function.

2> The default value of static variable is zero.

3> life time of static variable are till the termination of program.

# Keyword in C++ programming language →  
Keyword is nothing but Reserved words, whose meaning is already defined in the compiler.

Note →

1) we can not use keyword as a variable name & Constant name.

Example → int for =:10;  
                  Keyword → (X)

2) Keyword must be in lower case.

Keyword list →

Auto	case	register	unsigned
break	catch	return	Volatile
for	char	short	
void	continue	signed	
this	default	sizeof	
do	double	static	
while	enum	struct	
delete	extern	switch	
class	float	template	
const	inline	this	
goto	new	throw	
int	operator	try	
long	private	typedef	
else	public	union	
If	protected	long	

## # Identifier in C++ programming language →

Identifier refers to the name that is used to identify Variable, function, class and so on.

Example: → int q = 10;

void fun();

class Demo → Identifier

}