

# Unit $\rightarrow$ 1

## 1.) # Program ...

19/05/25

A program is a set of instruction written in programming language.

Set of instruction : (i) Alphabet (a to z).  
(ii) Digit (0 to 9).  
(iii) specific symbol (+, -, x, etc).

## # Variable ...

A variable is a dataname that stores the values.

A variable just like a container, it stores the value.

Eg:- a, b, add, sub, x, y, z:

For example :- (i)  $\begin{matrix} a & & b & & & & \text{add} \\ \boxed{10} & + & \boxed{20} & = & \boxed{30} \\ \text{add} = a + b \end{matrix}$  This is a instruction.

(ii)  $\begin{matrix} x & & y & & & & \text{sub} \\ \boxed{20} & - & \boxed{10} & = & \boxed{10} \\ \text{Sub} = x - y \end{matrix}$  instruction.

★ What is programming language?

A programming language is a medium by which instruction are transmitted to the computer to perform some specific operation / calculation.

★ Programming ...

The process of written the program is called programming.

★ Programmer ...

The person who write the program is called programmer.

## # Generation of Programming Language ...

① 1st Generation Language ... (Machine language) <sup>binary language</sup> 0,1

- Machine dependent language :- Program run on one computer cannot be run on another computer.

eg:-  $6 = 110$   
 $A = 1000001$

Character → ASCII Code  
American Standard Code for Information Interchange.

A = 65

B = 66

Z = 90

a = 97

b = 98

z = 100

Decimal Format

② 2nd Generation Language ... (Assembly language)

- Machine dependent language.

Language translator

↓  
Assembler

e.g.:-

A, B = dup

ADD

SUB

MULT

(Symbol)

↓  
Mnemonics code

★ Assembler :- Assembler is a language translator that convert program written in Assembly language into Machine language.

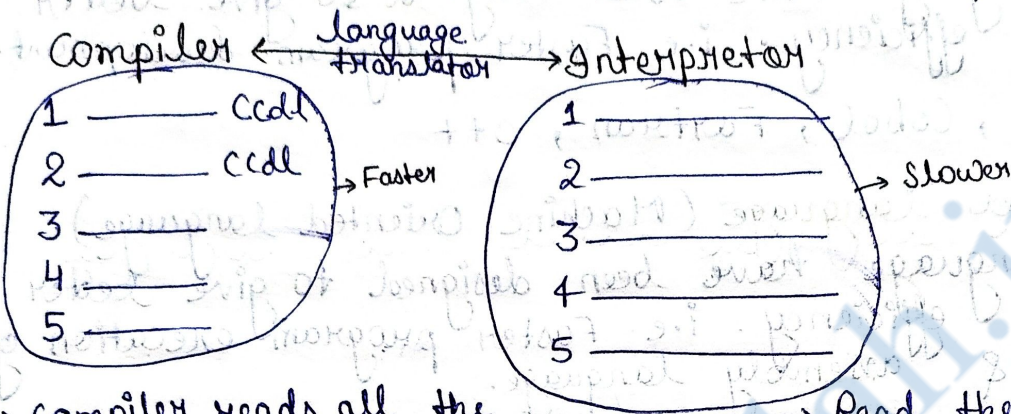
Machine language + Assembly language

↓  
Low level language

### ③ 3<sup>rd</sup> Generation Language ... (High level language and middle level language)

- Machine independent language
- Programs run on one computer can be run on another computer.

(We need language translator → Compiler  
→ Interpreter)



→ Compiler reads all the lines at a time & convert it into machine language.

→ Read the instruction line at a time & convert it into machine code and then read another line and so on.

Compiled Programming Language

Interpreted Programming Language

Compiler  
e.g.: C, C++, Cobol, Fortran, VC++

Interpreter  
e.g.: Python, JavaScript, VBScript etc...

Compiled & Interpreted Programming Language

Compiler & Interpreter

e.g.: Java

High level language → C++, Java, VC++, C#, Smalltalk, Simula-67, Python.

Middle level language → C, Cobol, Fortran etc.

C-Compiler → Turbo C, Dosbox, Turbo C++

★ Why C is called middle level language?

All the programming language can be divided into two categories:-

① High level language (Program Oriented language).  
→ These language have been designed to give better program efficiency. i.e. Faster program development.

Ex:- Basic, Cobol, Fortran, C++

② Low level language (Machine Oriented language).  
→ These language have been designed to give better machine efficiency. i.e. Faster program execution of machine & assembly language.

"C" falls in between both these categories that's why C is called middle level language. Since it has designed to give relatively better program efficiency & relatively better machine efficiency.

★ What is C?

→ C is a POP (Procedural Oriented Programming) developed by Dennis Ritchie in 1972 at Bell laboratories in USA at AT & T (American telegraph & Telecommunication).

★ Programming Techniques :-

Two types of programming techniques are commonly used:-

(i) POP (Procedural Oriented Programming) → C, Cobol, Fortran etc-

(ii) OOP (Object Oriented Programming)

↓  
C++, Java, VC++, C#, Python etc-

# # History of C-Program...

30/05/25

Friday

Algorithmic Language

ALGOL

Developed by International group in 1960

Basic combined programming language

BCPL

Developed by Martin Richards in 1967

B

Developed by Ken Thompson in 1970

Traditional C

Developed by Dennis Ritchie in 1972

K & RC

Developed by Kernighan & Ritchie in 1978

American National Standard Institute

ANSI C

Developed by ANSI Committee in 1989

International standard programming organizations

ANSI C in ISO C

Approved by ISO committee in 1990

C99

Approved by standard committee in 1992

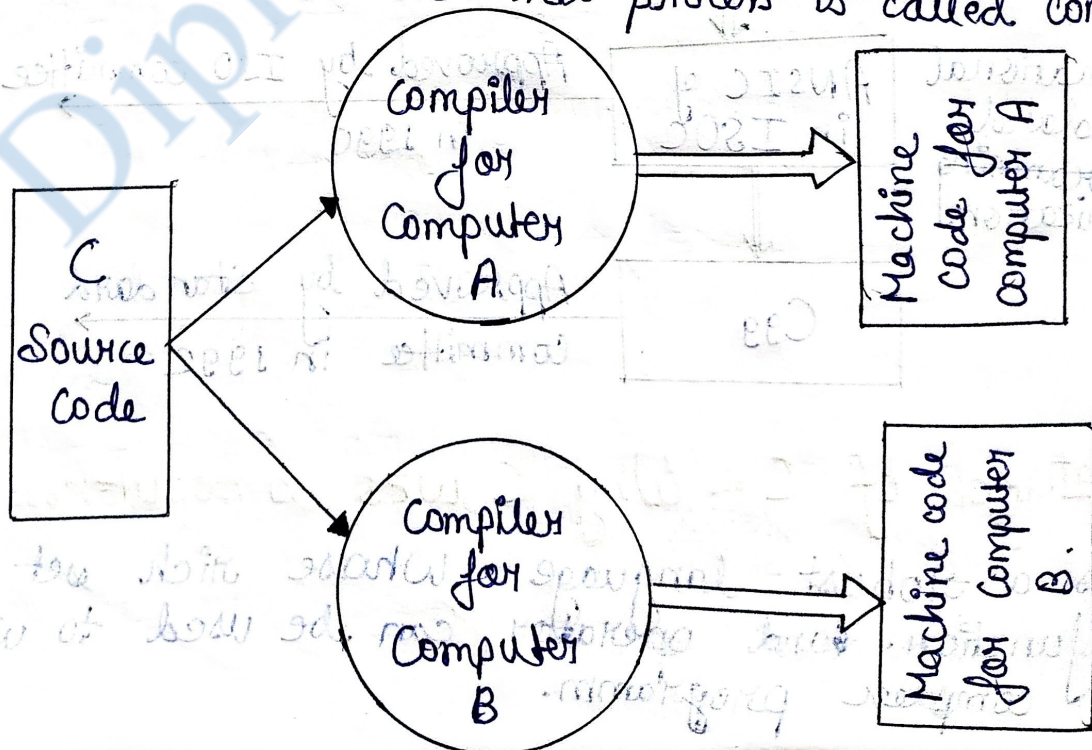
## # Features of C.../Why C was so popular...

- C is a <sup>powerful</sup> robust language whose rich set of built in function and operators can be used to write any complex program.

- C are fast in an efficient. This is due to its variety of data types and powerful operators. It is many times faster than basic.
- There are only 32 reserve keyword which can be used for developing programmes.
- C is highly portable this means that C programm written for one computer can be run on another computer.
- It can be compiled on a variety of computers.
- It is a structured language.
- Programm written in C has comparatively small in size.
- It has high level construct.
- It support pointer implementation.

## # Compilation Process

- A sequence of binary instructions consisting of 1 and 0 bits is called as machine code. High-level programming languages such as C, C++, Java, etc. consist of keywords that are closer to human languages. such as English.
- Hence, a program written in C (or any other high-level language) needs to be converted to its equivalent machine code. This process is called compilation.

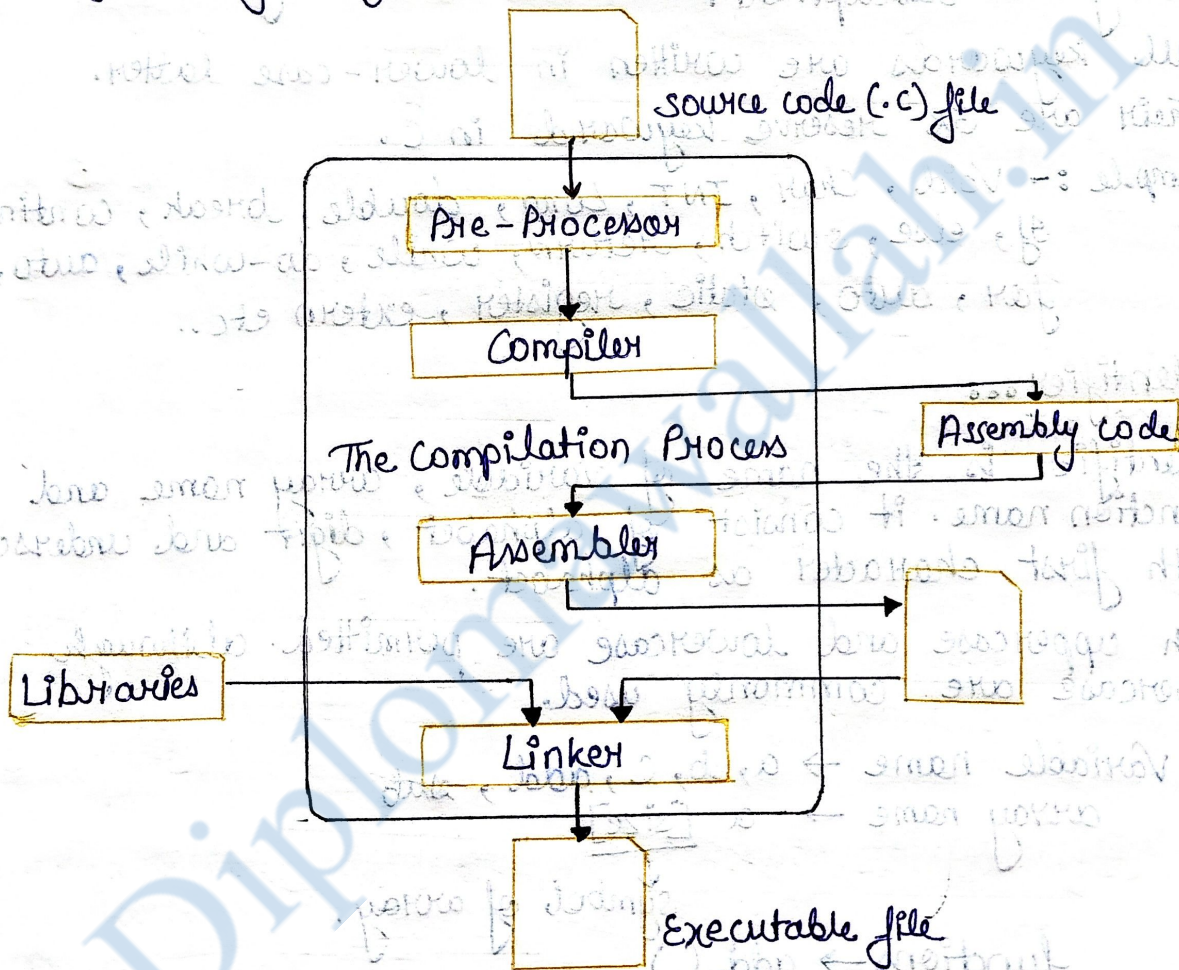


## ★ C Compilation Process steps ...

The Compilation Process has four different steps :-

- (i) Preprocessing
- (ii) Compiling
- (iii) Assembling
- (iv) Linking

The following diagram illustrate the compilation process.



## 2.) What is token / C tokens ...

- The smallest individual in it is a program is known as token. A program is made up of tokens.

There are six types of tokens :-

- (i) keywords / reserve keyword
- (ii) Identifier
- (iii) Literal / Constant

- (iv) String
- (v) Special Symbol
- (vi) Operator

(i) Keywords / Reserve keywords ...

- All keywords have fixed meaning and these meaning/keyword cannot be changed.
- The keyword serves as the basic building block in program development.
- All keywords are written in lower-case letter.
- There are 32 reserve keyword in C.

Example :- Void, Char, INT, Long, double, break, continue, if, else, switch, return, while, do-while, auto, for, auto, static, register, extern etc..

(ii) Identifier ...

- Identifier is the name of variable, array name and function name. it consist of alphabet, digit and underscore with first character as alphabet.
- Both uppercase and lowercase are permitted although lowercase are commonly used.

Variable name → a, b, c, add, sub  
 array name → a [size]

↓  
 symbol of array.

functions → add ( )

sub ( )

mult ( )

→ function

(iii) Variable ...

- A variable is a data name that stores a value. It is just like container which stores a value. A variable name consist of alphabet, digit or underscore or subject to the following condition.

## Conditions -

i) The first character must be alphabet than followed by digit or underscore.

Ex :- a1 - 1bc ✓  
1ab - bc x  
- 1bc x

ii) A variable name can be any length.

iii) White spaces are not allowed in the variable name.

Ex :- ab\_cd ✓  
ab cd x

iv) Reserve keyword cannot be used as variable name.

v) Both uppercase and lowercase character are different that's why C is called case-sensitive language.

Note : ① It is always good practice to only use alphabet as variable name, avoids digit and underscore.

② Maximum 3 to 4 used as variable.

# How to declare the Variable ?

→ Syntax :-

data type variable name = value ;

Note :- Without datatype variable name is meaningless.

## 3) Data types ...

A datatype is a way or method to identify the types of data and associated operation perform on that operation.

There are 4 fundamental or primary datatype on C :-

(i) char

(ii) int

(iii) float

(iv) Double

(i) Char ... (Character) ...

→ Only 1 character (character may be alphabet, digit or special symbol) enclose in a single quote mark (' ').

Example :- 'A', 'a', '5', '\$'.

- ① Memory occupied in RAM = 1 byte.
- ② Range of values = -128 to 127.
- ③ Control string = %c.
- ④ Control string → Represents the format specification to represent the value.

Char = %c.

02/06/25

Monday

(ii) Int ... (Integer) ...

→ Decimal number without fractional part with optional + or - sign.

eg :- 15, -20, -60 etc.

- ① Memory occupied in RAM = 2 byte
- ② Range of values = -32,768 to 32,767
- ③ Control string = %d

(iii) Float ... (Floating point Precision) ...

→ Decimal number with fractional part with optional + or - sign.

eg :- 12.25, -50.60.

3.4E-38 to 3.4E+38  
3.4 x 10<sup>-38</sup> to 3.4 x 10<sup>38</sup>

- ① Memory occupied in RAM = 4 byte.
- ② Range of value → 3.4E-38 to 3.4E+38
- ③ Control string → %f

(iv) Double ... (Double floating point precision)

→ Decimal number with fractional part with optional + or - sign.

eg :- 12.25 - 50.60

- ① Memory occupied RAM = 8 byte
- ② Range of value =  $1.7E-308$  to  $1.7E+308$
- ③ Control string = %F

$1.7E-308$  to  $1.7E+308$   
 $1.7 \times 10^{-308}$  to  $1.7 \times 10^{308}$

Note :- From programming point of view we use the following datatypes.

Data type	Memory occupied in RAM	Range	Control string	Example
① Char	1 byte	-128 to 127	%c	'A', '5', '&'
② Int	2 byte	-32768 to 32767	%d	15, 50
③ Float	4 byte	$3.4E-38$ to $3.4E+38$	%f	15.5, -5.05

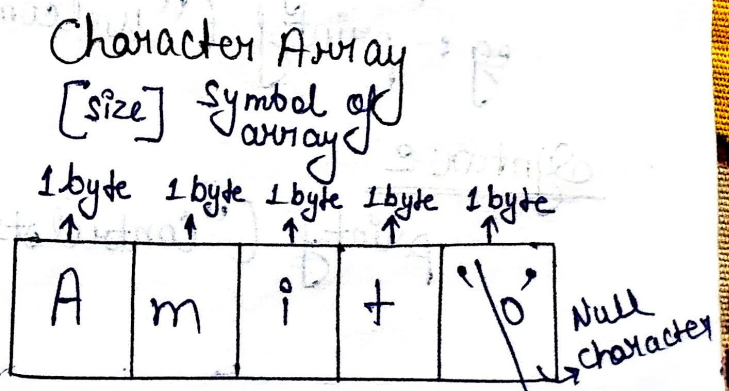
Derived data types:-

Additional we use another datatypes which is called derived data types:-

④ Character, Array/String :-

→ At least one character (character may be alphabet, digit or special symbol) inclosed in double quote mark (" ").

- eg:- "Amit"  
 "A"  
 "1995"  
 "&&&&"



→ In case of string the compiler automatically substitute Null character ('\\0') at the end of the name to indicate the name ends here.

Control string = %s

★ How to declare the variable....

Syntax 1 (Char / int / float)

datatype variablename = value

eg:- To declare 1 char value  
char x = 'p';

To declare 1 int value  
int a = 15;

To declare 1 float value  
float a = 15.5;

Syntax 2 (Character, Array / String)

char variablename [size] = value;

eg:- To declare 1 string value

char a [20] = "Amit";

# Syntax of printf ()

Syntax 1

printf (".....");

eg:- printf ("welcome to C");

Syntax 2

printf ("Control string", valuenam);

## Control string

char = %c

int = %d

float = %f

string = %s

eg:- Char a = 'p';  
printf ("%c", a);

P

## # Basic structure of C...

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main ()
```

```
{
```

```
Variable declaration;
```

```
char ch ();
```

```
[ Body of  
the program ]
```

```
getch ();
```

```
}
```

★ Write a program to declare 1 char value and display it :-

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main ()
```

```
{
```

```
Char a = 'p';
```

```
char ch ();
```

```
printf ("%c", a);
```

```
getch ();
```

```
}
```

output

p

```
printf ("charvalue = %c", a);
```

output

charvalue = p

\n = newline → It moves the cursor to the new line  
printf ("char value = %c\n", a);

output

```
char value = p-  
—
```

★ Write a program to declare 1 int value and display it.

```
#include <stdio.h>  
#include <conio.h>
```

```
void main ()
```

```
{
```

```
int a = 15;
```

```
clrscr ();
```

```
printf ("int value = %d\n", a);
```

```
getch ();
```

```
}
```

Output

```
int value = 15  
—
```

★ Write a program to declare 1 float value and display it.

```
#include <stdio.h>  
#include <conio.h>
```

```
void main ()
```

```
{
```

```
float a = 15.5;
```

```
clrscr ();
```

```
printf ("float value = %f\n", a);
```

```
getch ();
```

```
}
```

Output

```
float value = 15.5  
—
```

★ Write a program to declare 1 string value and display it.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char a [20] "Amit";
    clrscr();
    printf ("string value = %s\n", a);
    getch();
}
```

Output

String value = Amit

★ Write a program to take 1 char value, 1 int value, 1 float value and 1 string value.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char a = 'p';
    int b = 15;
    float c = 15.5;
    char a [20] "Amit";
    clrscr();
    printf ("char value = %c\n", a);
    printf ("int value = %d\n", b);
    printf ("float value = %f\n", c);
    printf ("string value = %s\n", a);
    getch();
}
```

Output

Char value = p  
int value = 15  
float value = 15.5  
String value = Amit

# scanf() → Input function

This function is used to interpreted values from keyboard.

\* Syntax of Input() function

```
scanf ("control string", &variablename);
```

Where control string...

char = %c

int = %d

float = %f

string = %s

& → Ampersand

The & (ampersand) tells where the variable is stored in memory.

Note: When value is input from the keyboard then how to declare variable.

Syntax:-

```
data type variablename;
```

char a;

int b;

float c;

char d [x];

\* Write a program to enter 1 char value using the keyboard and display it....

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main ( )
```

```
{
```

```
char a;
```

```
clrscr();
```

```

printf ("Enter char value =");
scanf ("%c", &a);
printf ("char value = %c\n", a);
getch ();
}

```

Output

```

Enter char value = P
char value = P

```

\* Write a program to enter 1 int value using the keyboard and display it.

```

#include <stdio.h>
#include <conio.h>
void main ()
{
int a;
clrscr ();
printf ("Enter int value =");
scanf ("%d", &a);
printf ("int value = %d\n", a);
getch ();
}

```

Output

```

Enter int value = 15
int value = 15

```

★ Write a program to enter 1 float value using the keyboard and display it.

```
#include <stdio.h>
#include <conio.h>
void main ( )
{
    float a;
    clrscr ( );
    printf ("Enter float value = ");
    scanf ("%f", &a);
    printf ("float value = %f\n", a);
    getch ( );
}
```

Output

```
Enter float value = 15.5
float value = 15.5
```

★ Write a program to enter 1 string value using the keyboard and display it.

```
#include <stdio.h>
#include <conio.h>
void main ( )
{
    char a [50];
    clrscr ( );
    printf ("Enter string value = ");
    scanf ("%s", &a);
    printf ("string value = %s\n", a);
    getch ( );
}
```

Output

```
Enter string value = Amit
String value = Amit
```

★ Write a program to enter 2 integer value using keyboard and display it.

```
#include <stdio.h>
#include <conio.h>
```

```
void main ( )
```

```
{
```

```
int a, b; → [ int a;
               int b; ]
```

```
clrscr ( );
```

```
printf ("Enter two int value =");
```

```
scanf ("%d %d", &a, &b);
```

```
printf ("a = %d\n", a);
```

```
printf ("b = %d\n", b);
```

```
getch ( );
```

```
}
```

```
printf ("a = %d\t", a);
printf ("b = %d\n", b);
```

Output

Enter two int value = 15 ↓ 20  
a = 15  
b = 20

a = 15      b = 20

\t → horizontal tab  
between two value

★ Write a program to enter one int and one float value using keyboard and display it.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
void main ( )
```

```
{
```

```
int a ;
```

```
float b ;
```

```
clrscr ( );
```

```
printf ("Enter one int & one float value =");
```

```
scanf ("%d %f", &a, &b);
```

```
printf ("a = %d\t", a);
```

```
printf ("b = %f\n", b);
```

```
getch ( );
```

```
}
```

Output

Enter one int and one float value =

★ Write a program to find the addition of two number.

```
#include <stdio.h>
#include <conio.h>
void main ( )
{
    int a, b, add;
    clrscr ( );
    printf ( "Enter value of a & b = " );
    scanf ( "%d %d", &a, &a );
    printf add = a + b;
    printf ( "Addition = %d \n", add );
    getch ( );
}
```

Output

Enter value of a & b = 5 ↓ 2

Addition = 7

—

★ Write a program to find the subtraction of two numbers.

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int a, b, sub;
    clrscr ();
    printf ("Enter value of a & b =");
    scanf ("%d %d", &a, &b);
    sub = a - b;
    printf ("subtract = %d\n", sub);
    getch ();
}
```

Output

Enter the value of a & b = 5 ↓ 2  
Subtract = 3

RAM  
space

★ Write a program to enter the radius of a circle. Find area of circle.

```

#include <stdio.h>
#include <conio.h>
void main (-)
{
    int r ;
    float area ;
    clrscr ();
    printf ("Enter the radius =");
    scanf ("%d", &r);
    area = 3.14 x r x r;
    printf ("Area = %f \n", area);
    getch ();
}

```

$$A = \pi r^2$$

\* Write a program to enter 3 number and find the average.

```

#include <stdio.h>
#include <conio.h>
void main ()
{
    int a, b, c;
    float avg;
    clrscr ();
    printf ("Enter value of a, b & c =");
    scanf ("%d %d %d", &a, &b, &c);
    avg = (a+b+c)/3.0;
    printf ("average = %f \n", avg);
    getch ();
}

```

$$\text{area of } \Delta = \frac{1}{2} \times b \times h$$

/(Divide) → Quotient (integer)

- ①  $\text{int}/\text{int} = \text{int} = 5/2 = 2$
- ②  $\text{int}/\text{float} = \text{float} = 5/2.0 = 2.5$
- ③  $\text{float}/\text{int} = \text{float} = 5.0/2 = 2.5$
- ④  $\text{float}/\text{float} = \text{float} = 5.0/2.0 = 2.5$

$a = 5$

$b = 4$

$c = 4$

$\text{float avg} = (a+b+c)/3;$

$= 4.00000$

$\text{float avg} = (a+b+c)/3.0;$

$= 4.33333$

21/07/25

Monday

2. There are six types of tokens :-

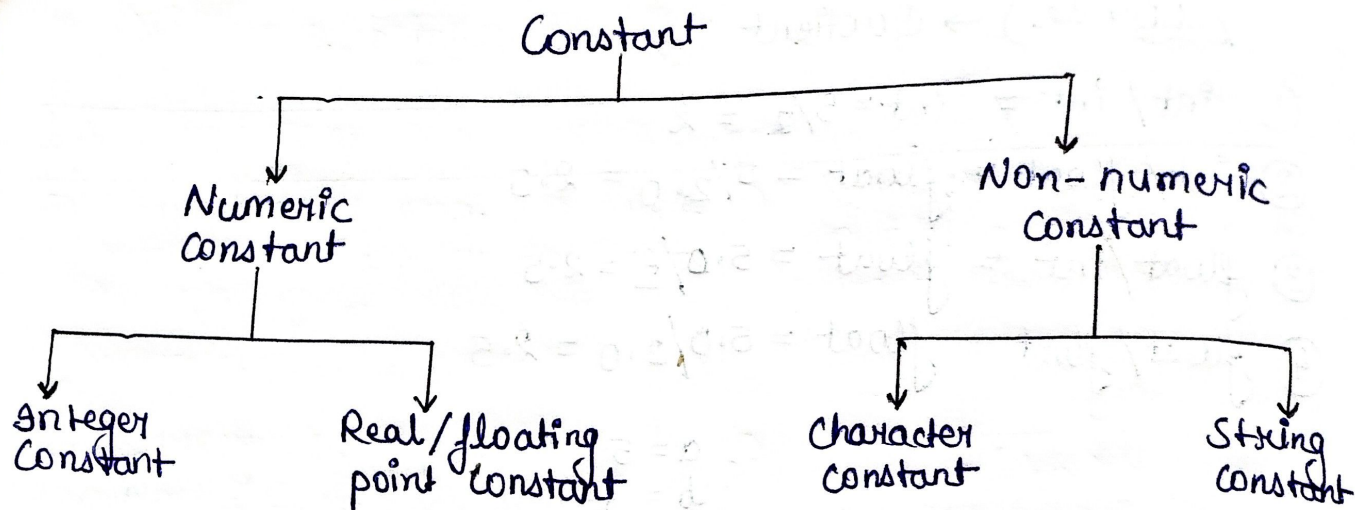
- (i) keywords / reserve key word
- (ii) Identifier
- (iii) Literal / constant
- (iv) String
- (v) Special Symbol
- (vi) Operator

(iii) **Literal / Constant** ...

→ Constant in C refers to the fixed value that do not change during the execution of the program.

→ C constant are classified into following categories :-





★ Integer Constant :-

- Decimal constant =  $\pm (0-9)$   
e.g  $\Rightarrow 1, 2, 3, -500$  control string  
%d
- Octal constant =  $(0-7)$  preceded by zero %.o  
e.g  $\rightarrow 0123 \quad 0456$
- Hexadecimal constant =  $(0-9)$  preceded by  $0x$  or  $0X$ .  
(10=A)  
(11=B) %.x  
(12=C)  
(13=D)  
(14=E)  
(15=F)

★ Real/floating point Constant :-

$\rightarrow$  It can be represented into two ways :-

① Decimal Notation (e.g  $\rightarrow 12.25, -500.60$ )

%.f

② Exponential Notation (e.g  $\rightarrow 1.2E25 \Rightarrow 1.2 \times 10^{25}$   
 $1.5e-30 \Rightarrow -1.5 \times 10^{-30}$ )

%.e

### \* Character Constant :-

Only one character (character may be alphabet, digit or special symbol) enclosed in single quote mark ( ' ' ).

e.g → 'A', 'a', '5', '&'

`%c`

### \* String Constant :-

At least one character (character may be alphabet, digit or special symbol), enclosed in double quote mark ( " " ).

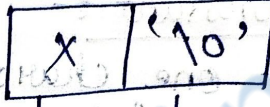
e.g → "Amit", "1995", "&&&&"

`%s`

In addition to the above constant there is another constant called backslash character constant.

Character	Meaning
① '\a'	audible ring (alert)
② '\t'	Horizontal Tab
③ '\v'	Vertical tab
④ '\n'	New line
⑤ '\0'	Null character
⑥ '\f'	Form feed etc.---

## # Difference between 'x' & "x".

'x'	"x"
<p>① Character enclosed in single quote is character constant so 'x' is a character constant.</p>	<p>① Character enclosed in double quote is string constant so, "x" is string constant.</p>
<p>② Memory occupied in RAM = 1 byte, so, 'x' takes 1 byte memory.</p>	<p>②   Memory occupied by it is 2 byte.</p>
<p>③ Control string = %c</p>	<p>③ Control string = %s</p>

### (iv) String ...

At least one character or sequence of character enclosed in double quote mark (" ") string is also known as character array.

eg → "Amit", "zero" etc.

Control string = %s

### \* Special Symbol :-

- ① ( ) → Parenthesis
- ② { } → Braces
- ③ [ ] → Bracket
- ④ < > → Angular Bracket

⑤ ^ → Caret

⑥ ~ → Tilde

⑦ : → Colon

⑧ ; → Semicolon

⑨

### (vi) Operator and Expression...

Expression :-

→ An expression is a combination of variable, constant and operator as per the syntax of the language.

eg:- ①  $x = (a+b)/15$

②  $y = (12+5-3)/12 * 10$

③  $z = (a+b)(c-d)$

Operator :-

→ An operator is a symbol that tells the computer to perform certain mathematical and logical operation.

addition	+
subtraction	-
multiplication	*
Division	/
Exponentiation	^

$2 + 3 = 5$   
 $5 - 2 = 3$   
 $2 * 3 = 6$   
 $6 / 2 = 3$   
 $2^3 = 8$