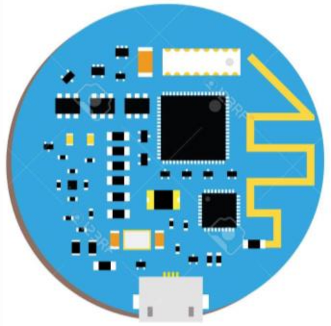
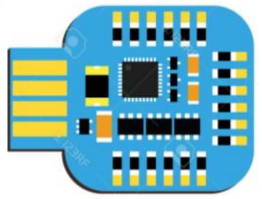


EMBEDDED C LANGUAGE - 2

Dr. Sarwan Singh

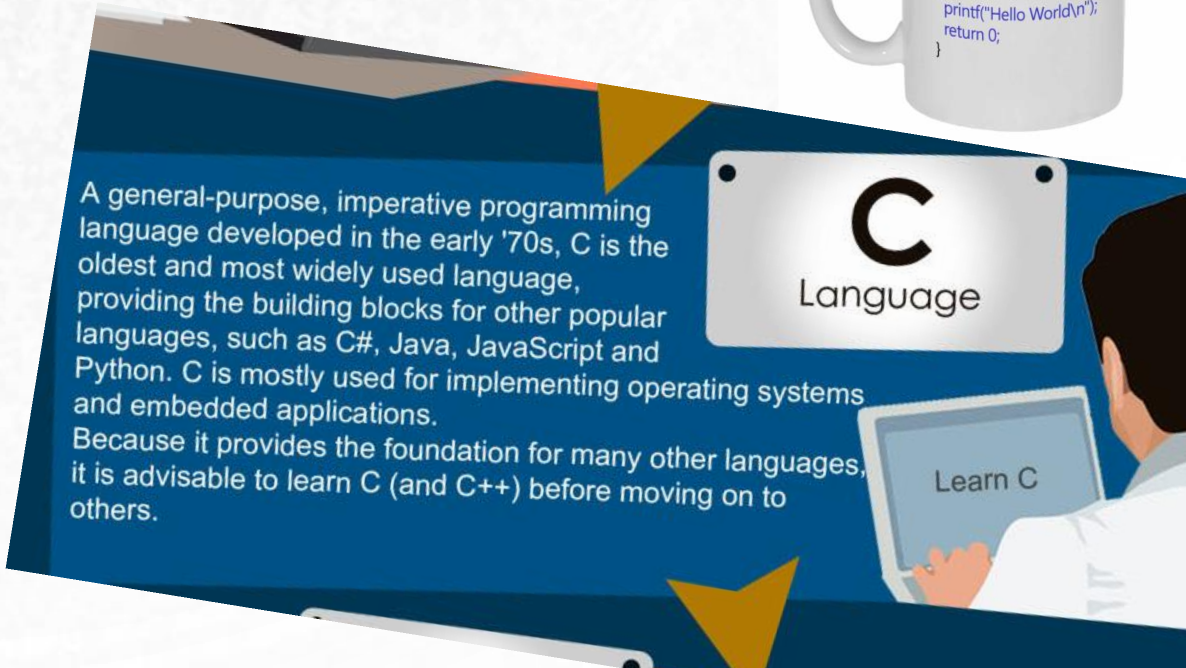
NIELIT Chandigarh



Agenda

- C language- data type, operators
- Flow and control statements
- Functions
- Header files

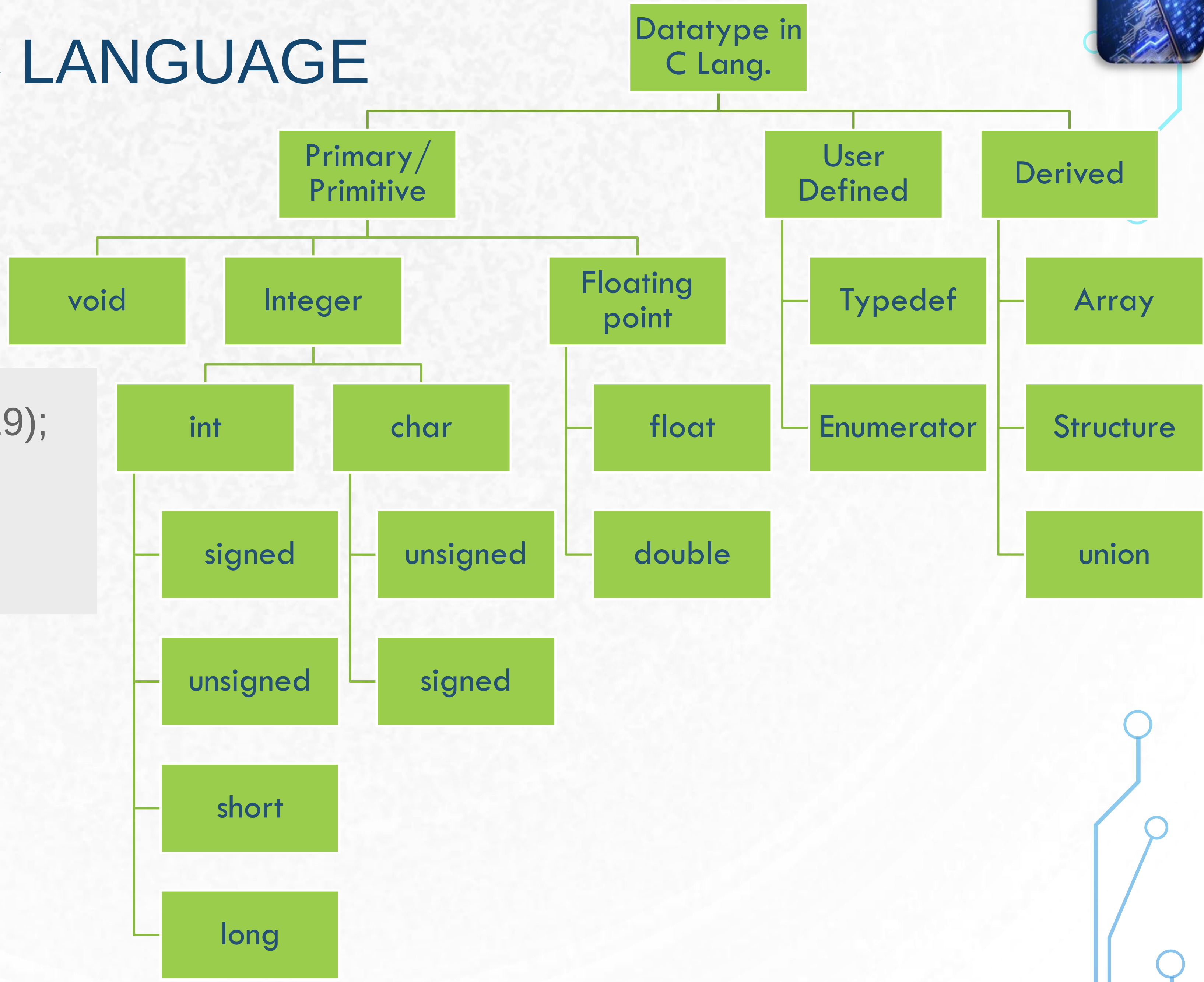
```
#include <stdio.h>

int main(void)
{
  printf("Hello World\n");
  return 0;
}
```


A general-purpose, imperative programming language developed in the early '70s, C is the oldest and most widely used language, providing the building blocks for other popular languages, such as C#, Java, JavaScript and Python. C is mostly used for implementing operating systems and embedded applications.
Because it provides the foundation for many other languages, it is advisable to learn C (and C++) before moving on to others.

DATATYPE IN C LANGUAGE

```
typedef int numbers;
numbers a=1;
```

```
enum prime (2,3,5,7,11,13,17,19);
enum prime a,b;
a=5;
b=19;
```



DATATYPE IN C LANGUAGE

Data type	Size	Range	Description
char	1 byte	-128 to 127	A character
signed char			
unsigned char	1 byte	0 to 255	A character
short	2 bytes	−32,767 to 32,767	Short signed integer of minimum 2 bytes
signed short			
signed short int			
unsigned short	2 bytes	0 to 65,535	Short unsigned integer of minimum 2 bytes
unsigned short int			
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647	An integer (Both positive as well as negative)
signed int			
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295	An unsigned integer (Positive integer)

DATATYPE IN C LANGUAGE

Data type	Size	Range	Description
long	4 bytes	-2,147,483,648 to 2,147,483,647	Long signed integer of minimum 4 bytes
signed long			
signed long int			
unsigned long	4 bytes	0 to 4,294,967,295	Long unsigned integer of minimum 4 bytes
unsigned long int			
float	4 bytes	1.2E-38 to 3.4E+38	Single precision floating point number
double	8 bytes	2.3E-308 to 1.7E+308	Double precision floating point number
long double	12 bytes	3.4E-4932 to 1.1E+4932	Double precision floating point number

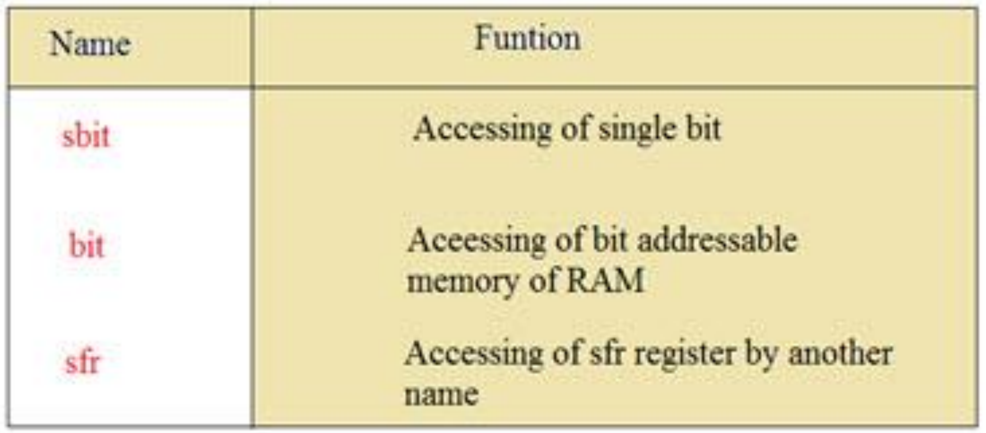
QUALIFIER-MODIFIER

- **register-** Local variable are stored in register instead of RAM
- **static** defined local variables do not lose their value between function calls.
- **typedef** used to create new type
- **extern** used to declare global variable
- **volatile** variable values might keep on changing without any explicit assignment by the program

Group	Qualifiers (Modifier)	Default Qualifiers (Modifier)
1	auto, register, static, extern, typedef	auto
2	signed, unsigned	signed
3	Short, long	Not Short, not long
4	Const	Not Const
5	Volatile	Not Volatile

EMBEDDED C DATATYPES

- **sbit:** This data type is used in case of accessing a single bit of SFR register.
  - sbit a=P2^1;
- **Bit**: This data type is used for accessing the bit addressable memory of RAM (20h-2fh).
  - bit c;
- **SFR:** This data type is used for accessing a SFR register by another name. All the SFR registers must be declared with capital letters.
  - SFR port0=0x80;



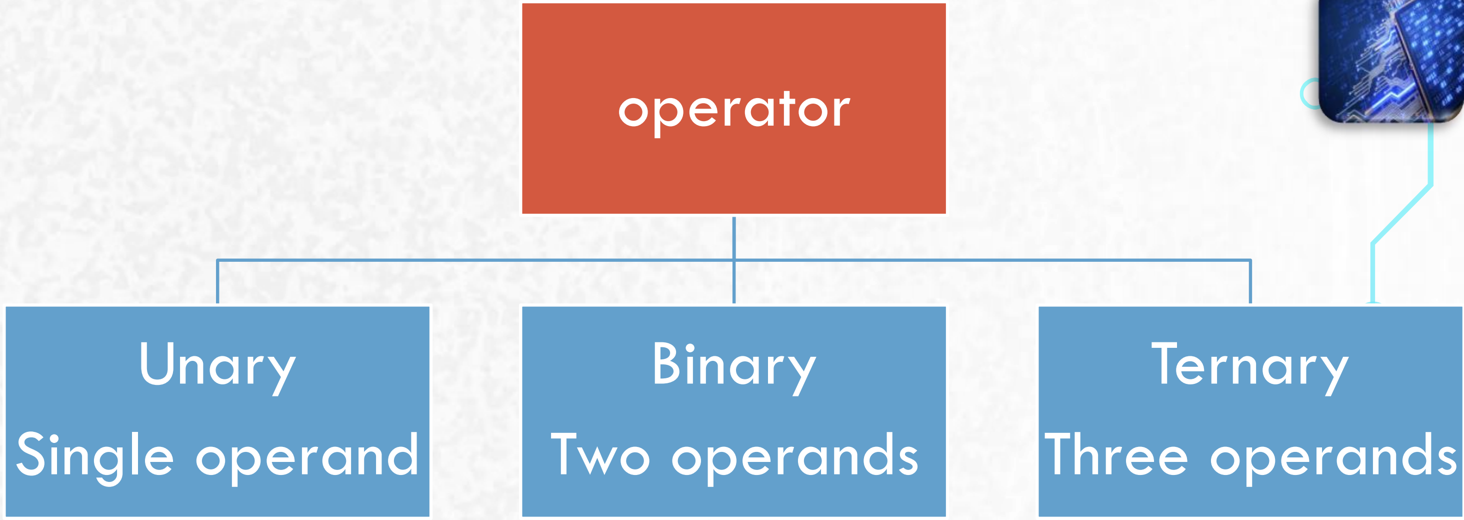
Name	Funtion
sbit	Accessing of single bit
bit	Accessing of bit addressable memory of RAM
sfr	Accessing of sfr register by another name

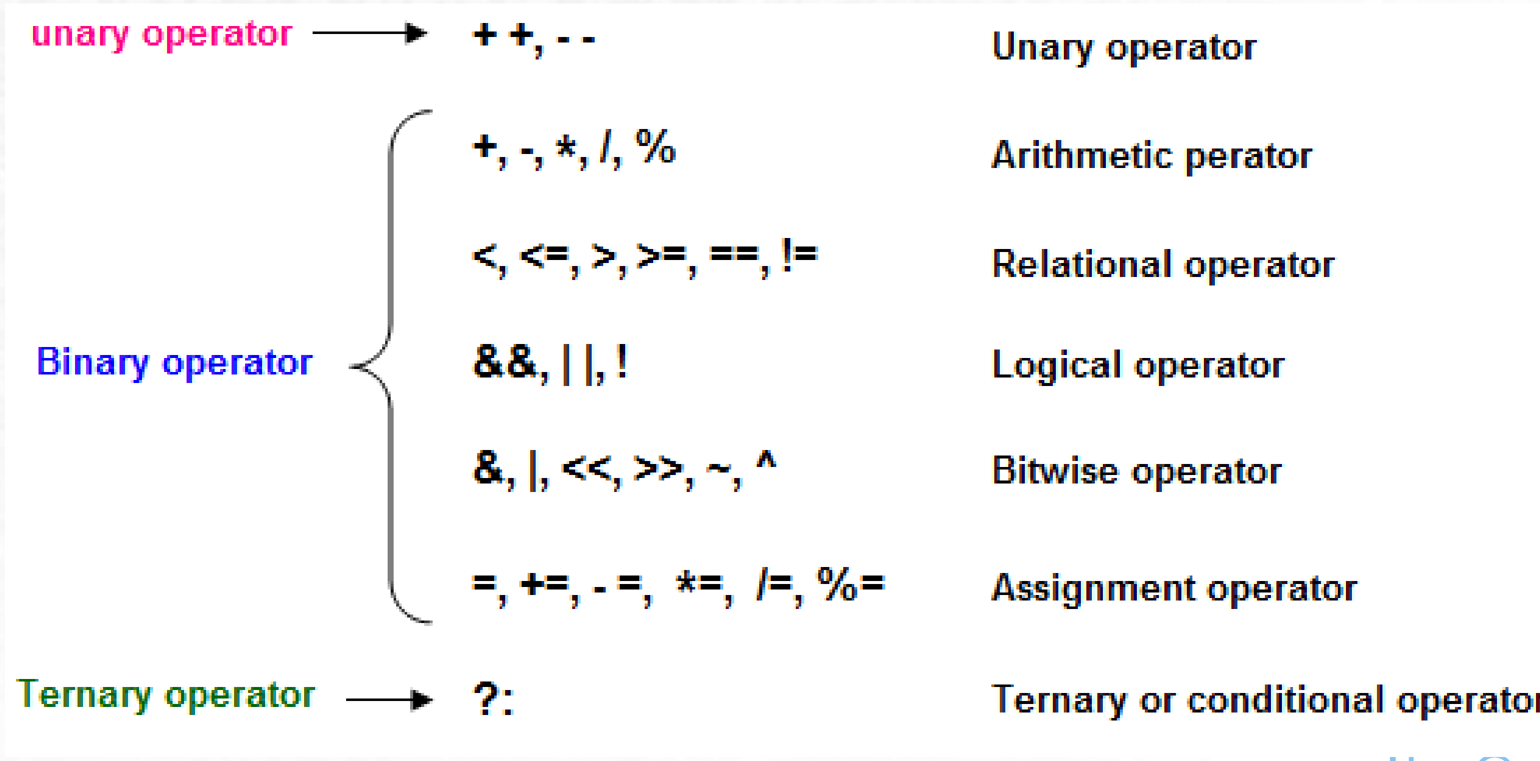
SFR Register: The SFR stands for **'Special Function Register'.** Microcontroller 8051 has 256 bytes of RAM memory.
This RAM is divided into two parts:
- the first part of 128 bytes is used for data storage, and
- the other of 128 bytes is used for SFR registers.

All peripheral devices like I/O ports, timers and counters are stored in the SFR register, and each element has a unique address.

Source:elprocus.com

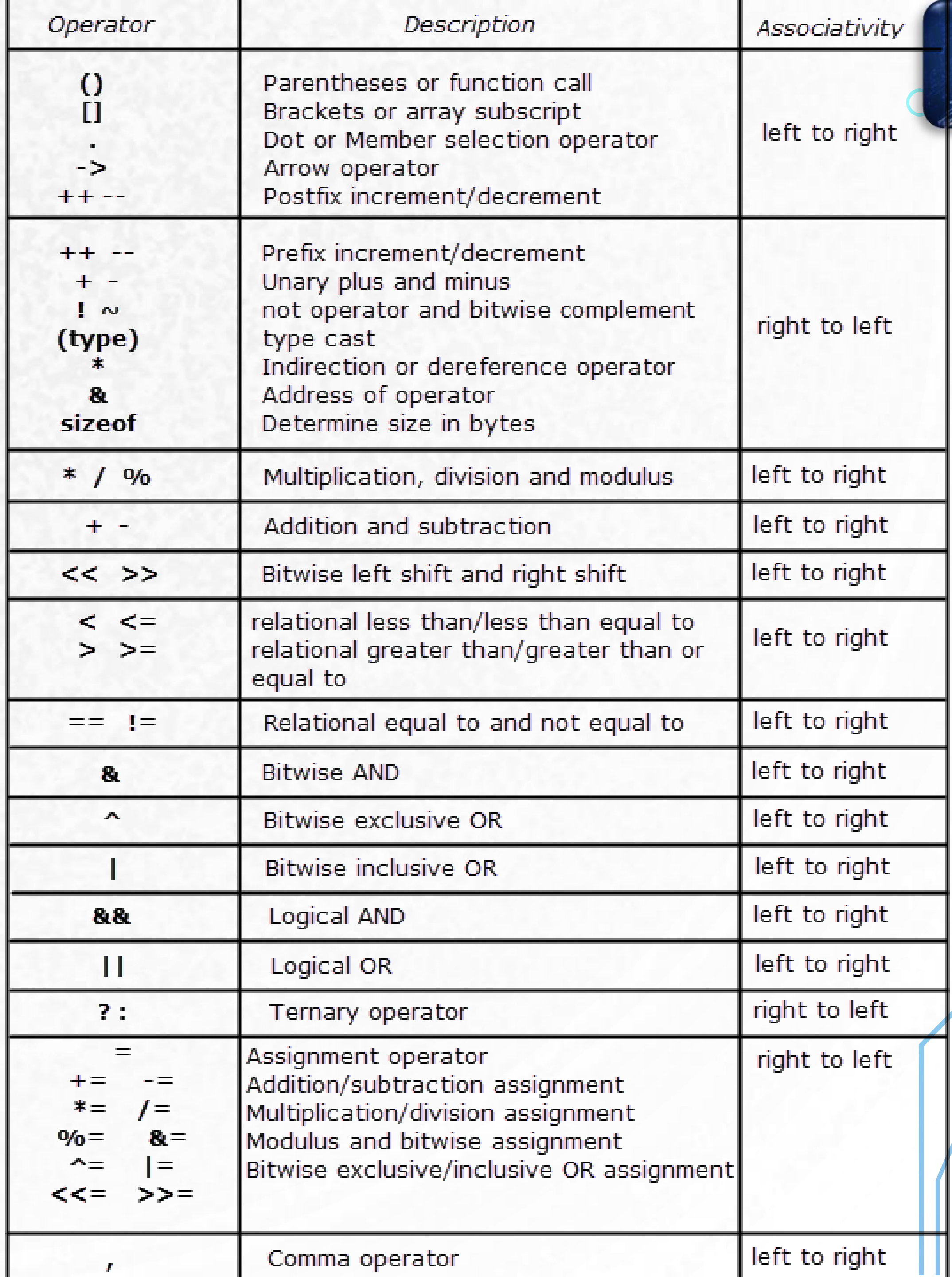
OPERATOR





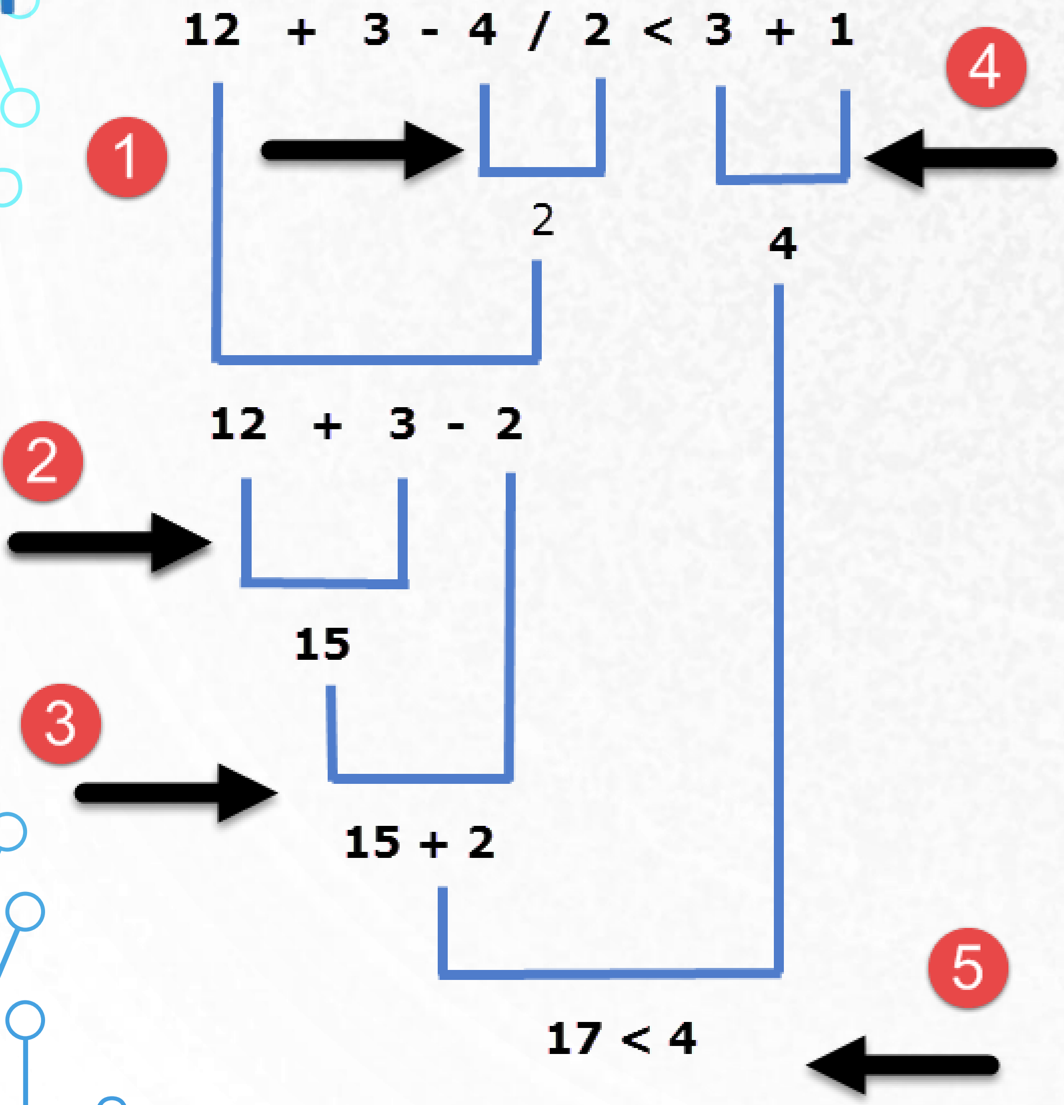
unary operator →	+ +, - -	Unary operator
Binary operator	+, -, *, /, %	Arithmetic perator
	<, <=, >, >=, ==, !=	Relational operator
	&&, | |, !	Logical operator
	&, |, <<, >>, ~, ^	Bitwise operator
	=, +=, - =, *=, /=, %=	Assignment operator
Ternary operator →	?:	Ternary or conditional operato

PRECEDENCE & ASSOCIATIVITY



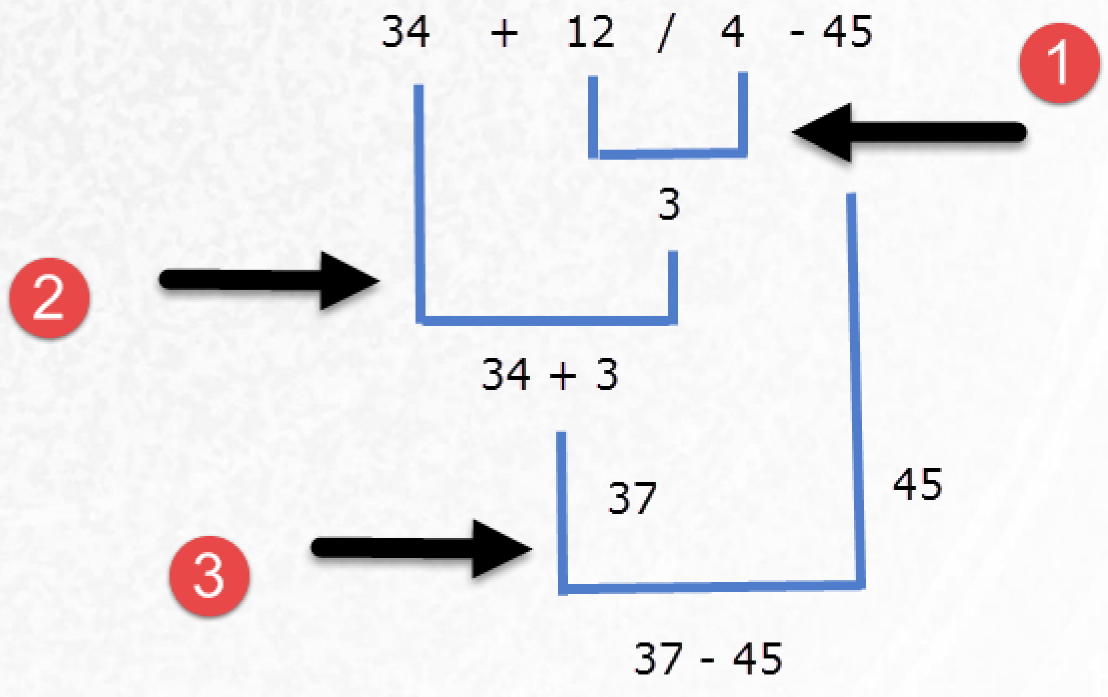
Operator	Description	Associativity
() [] . -> ++ --	Parentheses or function call Brackets or array subscript Dot or Member selection operator Arrow operator Postfix increment/decrement	left to right
++ -- + - ! ~ (type) * & sizeof	Prefix increment/decrement Unary plus and minus not operator and bitwise complement type cast Indirection or dereference operator Address of operator Determine size in bytes	right to left
* / %	Multiplication, division and modulus	left to right
+ -	Addition and subtraction	left to right
<< >>	Bitwise left shift and right shift	left to right
< <= > >=	relational less than/less than equal to relational greater than/greater than or equal to	left to right
== !=	Relational equal to and not equal to	left to right
&	Bitwise AND	left to right
^	Bitwise exclusive OR	left to right
|	Bitwise inclusive OR	left to right
&&	Logical AND	left to right
||	Logical OR	left to right
? :	Ternary operator	right to left
= += -= *= /= %= &= ^= |= <<= >>=	Assignment operator Addition/subtraction assignment Multiplication/division assignment Modulus and bitwise assignment Bitwise exclusive/inclusive OR assignment	right to left
,	Comma operator	left to right

$= 12 + 3 - 4 / 2 < 3 + 1$



Ans: 0

$= 34 + 12/4 - 45$



Ans : -8

BITWISE OPERATOR

- Bitwise AND operator &
- Bitwise OR operator |
- Bitwise XOR operator ^
- Bitwise complement operator ~
- Bitwise left shift operator <<
- Bitwise right shift operator >>

BITWISE OPERATOR

Write a C program using bitwise operator

- to check Least Significant Bit (LSB) / Most Significant Bit (MSB) of a number is set or not.
- to get / set nth bit of a number.
- to clear nth bit of a number.
- to toggle nth bit of a number.
- to get highest / lowest set bit of a number.
- to count trailing / leading zeros in a binary number.
- to flip bits of a binary number using bitwise operator.
- to count total zeros and ones in a binary number.
- to convert decimal to binary number system using bitwise operator.
- to swap two numbers using bitwise operator.
- check whether a number is even or odd using bitwise operator.

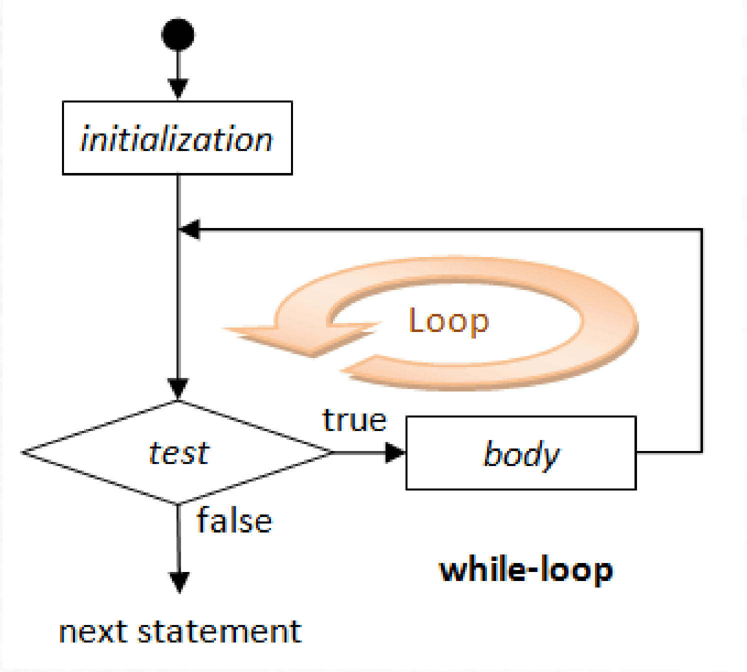
LOOP – *to perform repetitive task*

```
initialization-statement;
while (test) {
   loop-body;
}
next-statement;
```

<u>Other variants </u>:

do {   …. } while (condition)

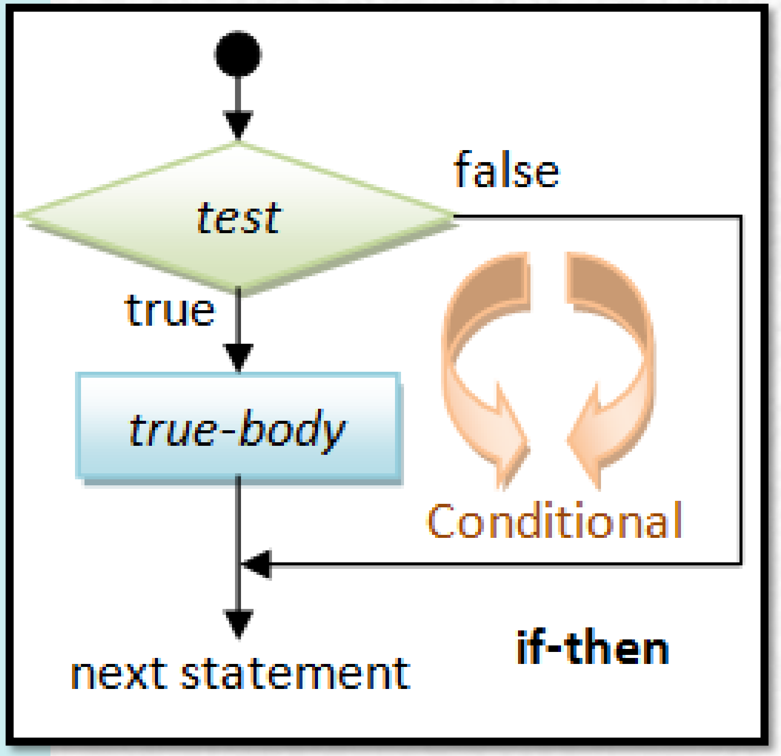
for(initialization ; condition ; inc/dec) { ….}

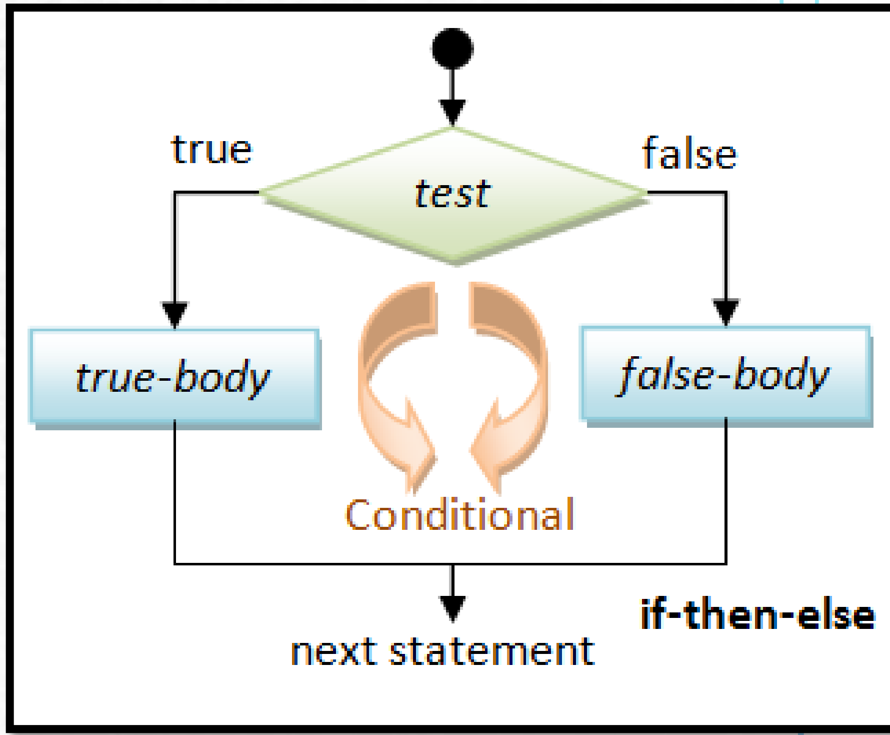


Source:ntu.edu.sg

CONTROL STATEMENT - ***conditional (or decision)***

```
// if-then
if ( test )
{ true-body; }
// if-then-else
if ( test )
{ true-body; }
else
{ false-body; }
```





Other variant

- switch case

FUNCTION / METHOD

- Reduces code duplication
- Repetitive task can be represented in form of method
- Make code modular
- provide abstraction

```
// A function that takes two integers as
// parameters and returns an integer
int max(int, int);

// A function that takes a int pointer and an
//int variable as parameters
// and returns an integer of type int
int *swap(int*,int);

// A function that takes a char and an int as
// parameters and returns an integer
int fun(char, int);
```

QUESTION

- Write a one line C function to round floating point numbers

```
int roundNo(float num)
{
    return num < 0 ? num - 0.5 : num + 0.5;
}
```

HEADER FILES

- Header file is a file that contains function declaration and macro definition for C in-built library functions.
- All C standard library functions are declared in many header files which are saved as file_name.h.