

C Language

-What is Dev C++ & Download, install Dev C++

-Dev-C++ is a free full-featured integrated development environment (IDE) for programming in C and C++.

-What is Syntax

Syntax is a set of rules that defines the structure of a language.

-What is Compiler

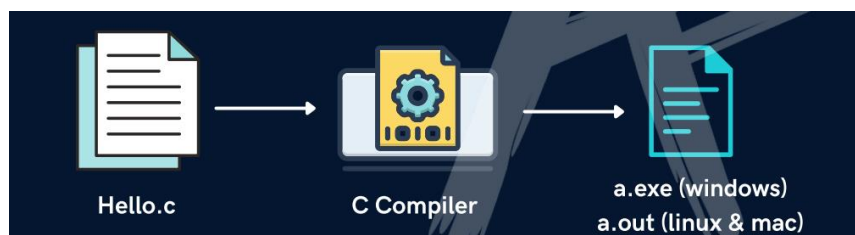
Compilers convert programming languages like Java or C++ into binary code that computers can understand. If the syntax is incorrect, the code will not compile.

Interpreters execute programming languages such as JavaScript or Python at runtime. The incorrect syntax will cause the code to fail.

-Demonstrate first program to print message and to show comments and basic errors.

-Compile the program n executable file.

Compilation



-What is Comment in Program

Lines that are not part of the program. E.g program definition

Single Line //

Multiple Lines /*
 */

-

-Output in C

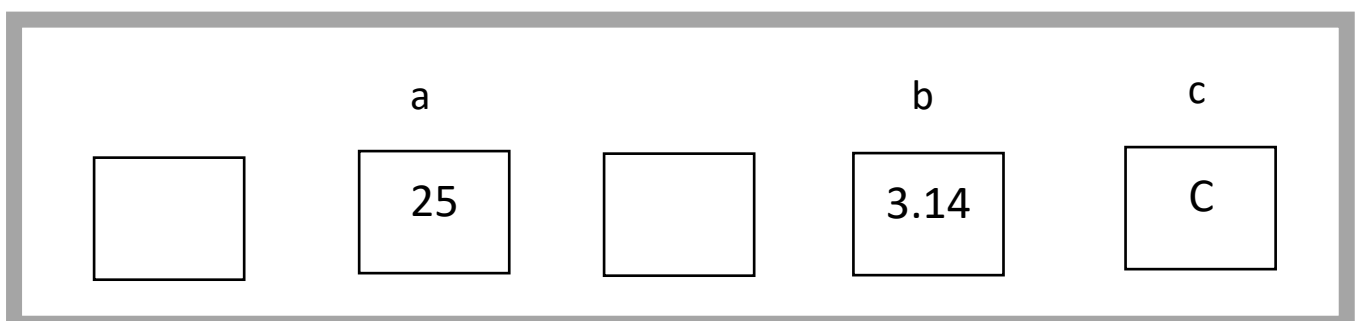
```
printf(" Hello C");  
printf(" \n Hello C"); \n for new line.  
printf(" \t Hello C"); \t for tab space.
```

Input in C

```
scanf("%d", &age); // age is an integer type of variable.
```

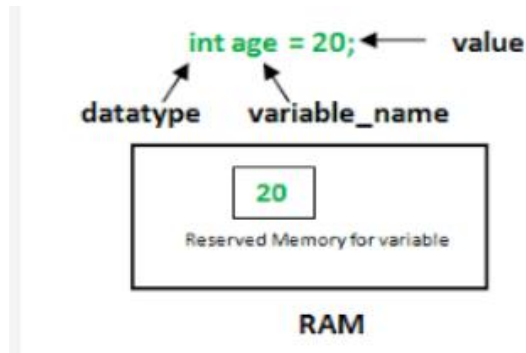
-What is Variable & Rules for the Variable

Variable is the name of a memory location which stores some data.



➤ Rules :

- Variables are case sensitive.
- 1st character is alphabet or '_'.
No other symbol other than '_'.
no comma/blank space

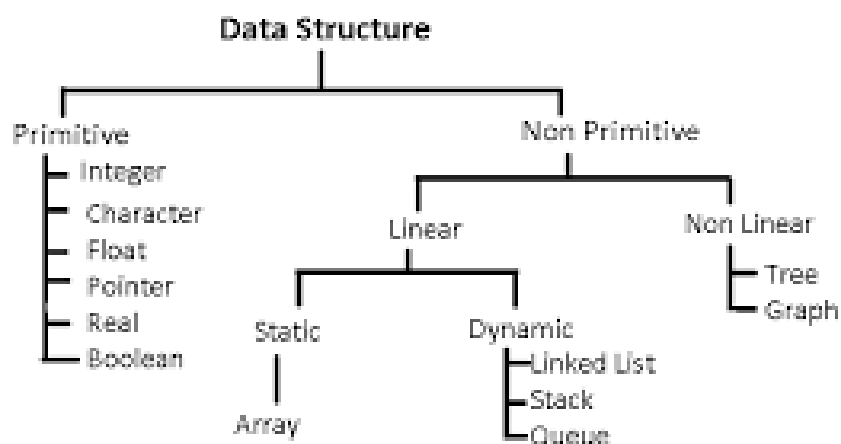


-Demonstrate the prog for user input and print the values.

-Demonstrate datatypes: int, float, long int, double, char.

-What is Data Structure & What is Data type

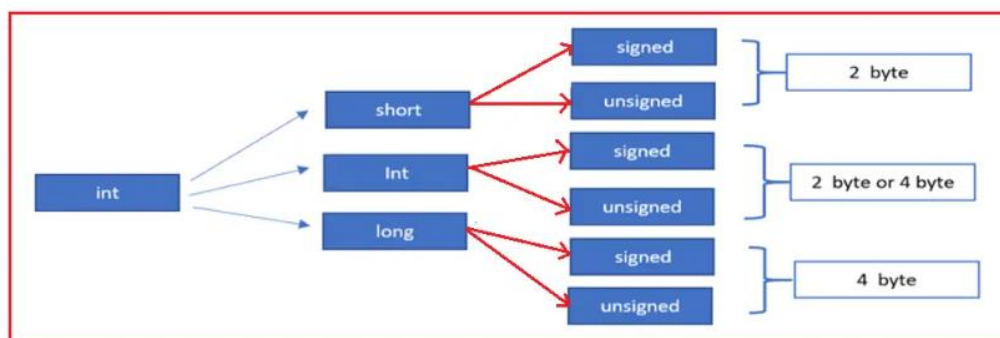
A data structure is a particular way of organizing data in a computer so that it can be used effectively.

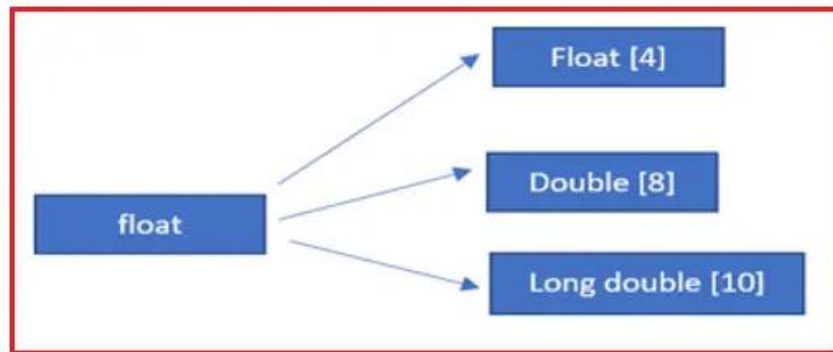
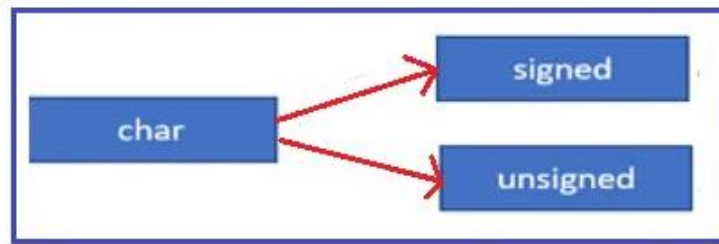


Data type : A type of data that a variable can store like int, float, char.

Data type	Size in bytes
Char or signed char	1
Unsigned char	1
int or signed int	2
Unsigned int	2
Short int or Unsigned short int	2
Signed short int	2
Long int or Signed long int	4
Unsigned long int	4
float	4
double	8
Long double	10

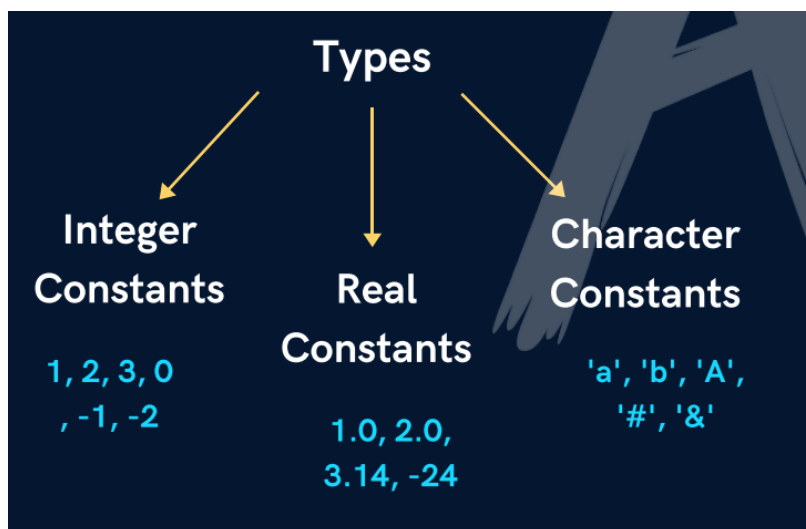
Data Types	Memory Size	Range
short int	2 byte	-32,768 to 32,767
signed short int	2 byte	-32,768 to 32,767
unsigned short int	2 byte	0 to 65,535
long int	4 byte	-2,147,483,648 to 2,147,483,647
signed long int	4 byte	-2,147,483,648 to 2,147,483,647
unsigned long int	4 byte	0 to 4,294,967,295
float	4 byte	
double	8 byte	
long double	10 byte	





-What is Constant

Values that don't change(fixed)



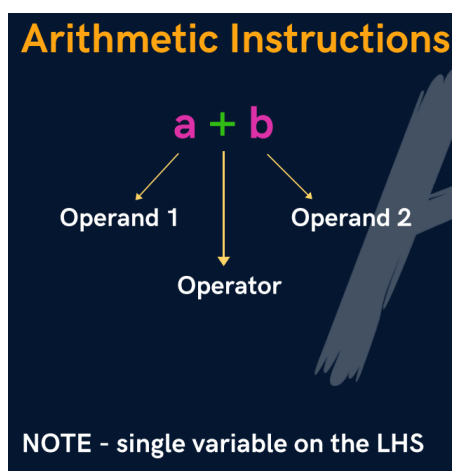
-Keywords

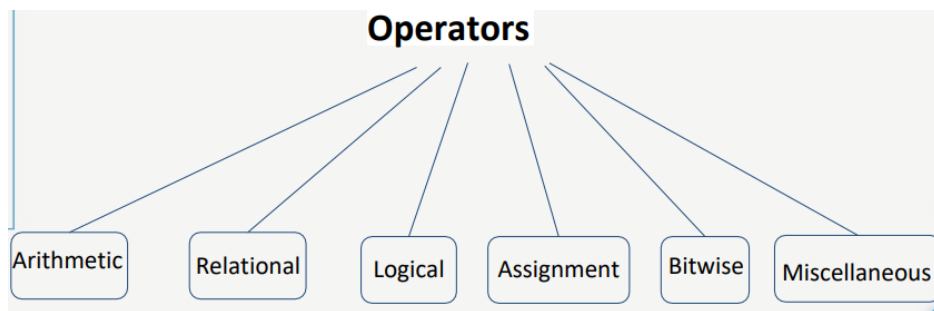
Reserved words that have special meaning to the compiler. 32 keywords in C.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned

-Operators

A symbol that takes one or more operands such as variables, expressions or values and operates on them to give an output.





Arithmetic Operators

Operator	Function	Example
+	Addition	var=a+b
-	Subtraction	var=a-b
*	Multiplication	var=a*b
/	Division	var=a/b
%	Modulo	var=a%b
++	Increment	var++
--	Decrement	var--

Relational Operators

Operator	Meaning of Operator	Example
==	Equal to	5 == 3 is evaluated to 0
>	Greater than	5 > 3 is evaluated to 1
<	Less than	5 < 3 is evaluated to 0
!=	Not equal to	5 != 3 is evaluated to 1
>=	Greater than or equal to	5 >= 3 is evaluated to 1
<=	Less than or equal to	5 <= 3 is evaluated to 0

Logical Operators

Operator	Meaning	Example
&&	Logical AND. True only if all operands are true	If c = 5 and d = 2 then, expression ((c==5) && (d>5)) equals to 0.
	Logical OR. True only if either one operand is true	If c = 5 and d = 2 then, expression ((c==5) (d>5)) equals to 1.
!	Logical NOT. True only if the operand is 0	If c = 5 then, expression !(c==5) equals to 0.

Bitwise Operators

Operators	Meaning of operators
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left
>>	Shift right

Miscellaneous Operators

Operator	Description	Example
sizeof()	Returns the size of a variable.	sizeof(a), where a is integer, will return 4.
&	Returns the address of a variable.	&a; returns the actual address of the variable.
*	Pointer to a variable.	*a;
? :	Conditional Expression.	If Condition is true ? then value X : otherwise value ..

-Different types of conditional statements :

- 1).Simple if statement (without using else statement)
- 2).if .. else statement
- 3)..Nested if.. else statement
- 4).else if ladder

1).Simple if statement (without using else statement)

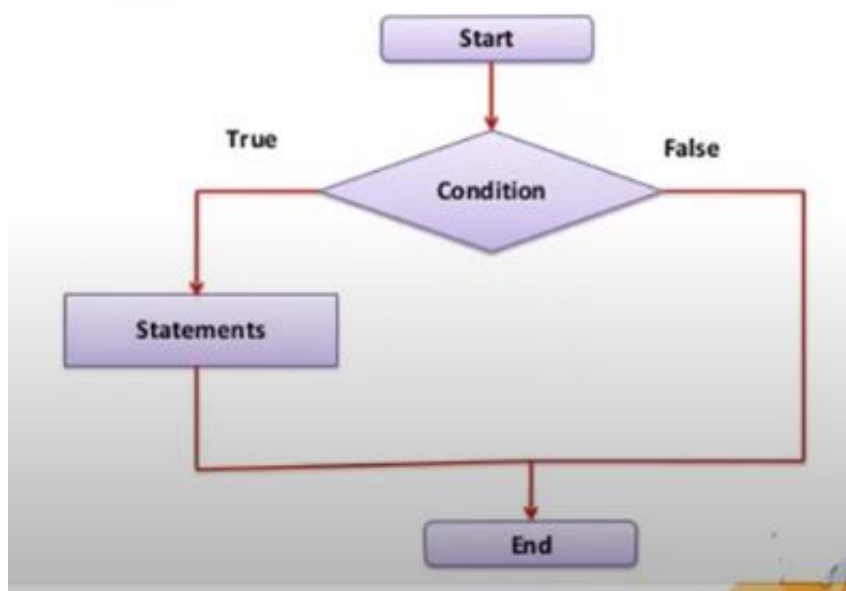
if (condition)

{

statements;

... ..

}



```
//C Program of absolute number conversion.
```

```
void main( )  
{  
    int num;  
    printf("\n Enter any number : ");  
    scanf("%d",&num);  
    if (num < 0)  
    {  
        num = num * (-1) ;  
    }  
    printf( "\n Absolute value is : %d",num);  
    getch( );  
}
```

out put of the program :

Enter any number : -25

Absolute value of enter number is : 25

Other Examples :

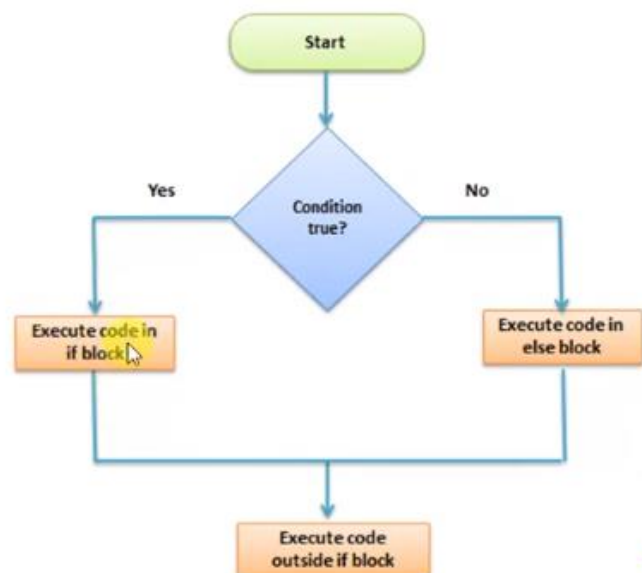
- If age of person > 18 then eligible for voting.
- If salesman is selling >50000 then 10% incentive
- If purchase is >1000 then 25% discount
- If recharge is > 399 then eligible for cash back offer.

Dr. Ashwin R. Dobariya

2).if .. else statement

General syntax is :

```
If ( Test condition )  
{  
    Statement block-1;  
}  
else  
{  
    Statement block-2;  
}
```



// C Program to check odd or even number.

```
void main( )  
{   int x;  
    printf("\n Enter any number : ");  
    scanf("%d",&x);  
    if ( x % 2 == 0)  
        printf("\n Input number is Even ");  
    else  
        printf("\n Input number is Odd");  
    getch ( );  
}
```

Output of the program :

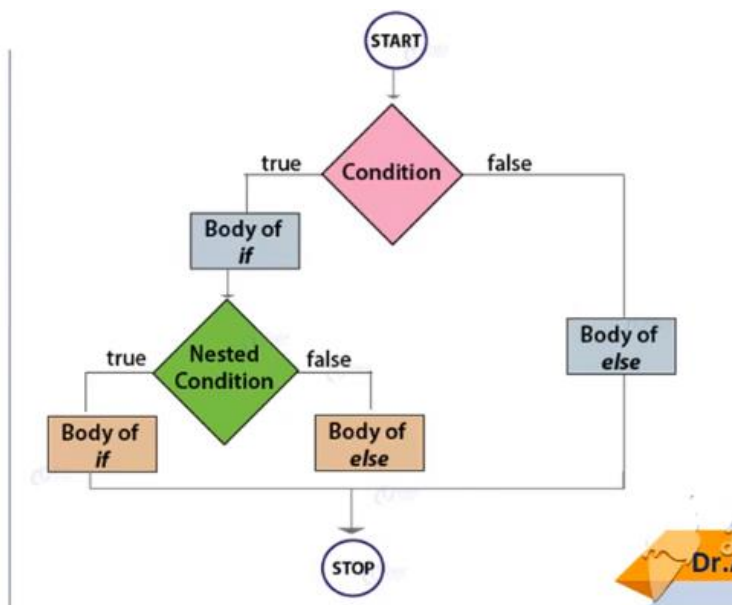
Enter any number : 7
Input number is Odd

Enter any number : 8
Input number is Even

3)..Nested if.. else statement

General Syntax :

```
If( Test condition )  
{  
    If( Test condition )  
        Statement block;  
    else  
        Statement block;  
}  
else  
{  
    if (Test condition)  
        Statement block;  
    else  
        Statement block;
```



```
void main( )
```

```
{
```

```
float P=75;
```

```
if(P >= 70)
```

```
printf("\n Your class is Distinction");
```

```
else
```

```
{
```

```
if(P >= 60)
```

```
printf("\n First Class");
```

```
else
```

```
if( P >=50)
```

```
printf("\n Second Class");
```

```
else
```

```
if ( P>=40)
```

```
printf("\n Pass Class");
```

```
else
```

```
printf("\n Result is Fail");
```

Output of the program :

Enter your percentage : 75

Your class is Distinction

4).else if ladder

```
if( Test condition )
```

```
statement-1;
```

```
else if(Test condition)
```

```
Statement-2;
```

```
else if (Test condition)
```

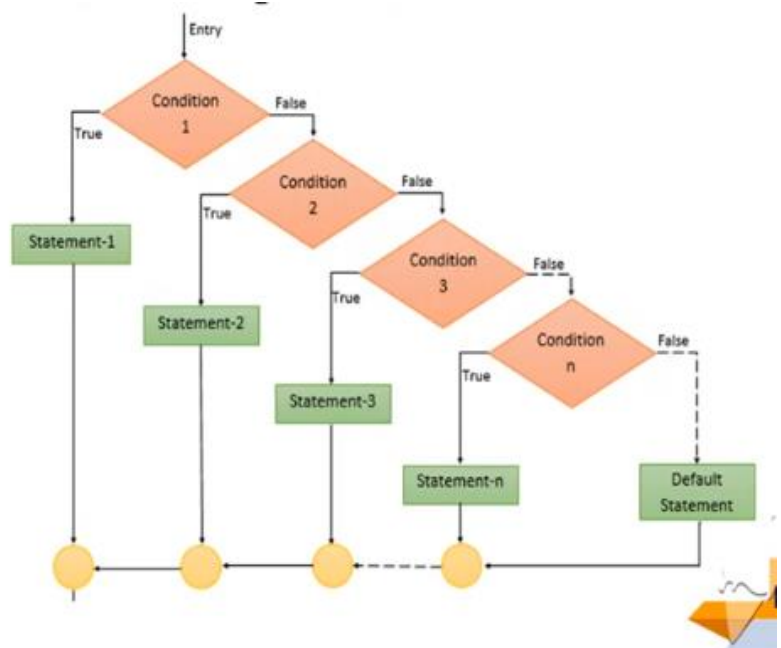
```
Statement-3;
```

```
else if(Test condition)
```

```
statement-4;
```

```
else
```

```
statement-5;
```



```

void main( )
{
    float P;
    if(P >= 70)
        printf("\n Your class is Distinction");
    else If(P >= 60)
        printf("\n First Class");
    else if( P >=50)
        printf("\n Second Class");
    else If ( P>=40)
        printf("\n Pass Class");
    else
        printf("\n Result is Fail");
}

```

Generally this Escape sequence character is used to display output in printf () function.

No.	Escape sequence character	Use
1	\n	a new line character
2	\t	Horizontal tab
3	\a	to generate sound (bell)
4	\b	Back space character
5	\r	carriage return
6	\v	Vertical tab
7	\"	double quote
8	'	single quote
9	\\	single backslash
10	'\0'	NULL

Ternery op

Condition?true:false