

# C Language

# 01 C Language Introduction

- ◇ C Language introduction
- ◇ About C Language

## C Language introduction

C Language is a programming language.

C Language is used for creating system application which directly interact with the hardware(h\w) devices.

## About C Language

System programming language.

Structured programming language.

Mother language.

Mid-level language

Note : Because it has features of both assembly level language i.e Low level languages and high level languages.

# 02 history of C Language

## ◇ History of C

### History of C

C Language is developed by Dennis Ritchie [1972 at bell laboratories].

Language	Year	Developer
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional	1972	Dennis Ritchie
K & RC	1978	Kernighan & Dennis Ritchie
ANSIC	1989	ANSI Commitee
ANSI/ISOC	1990	ISO Commitee

# 03 Operator in C

- ◆ Operators
- ◆ Types of operators

## Operators

Operator are one of the features in C which has symbols that can be used to perform mathematical , relational , bitwise , conditional , or logical manipulations.

## Types of operator

There are four types operators

Arithmetic operators [ + , - , \* , / , % , ++ , -- ].

Comparison operator [ == , != , > , < , >= , <= ].

Logical operator [ && , || , ! ].

Bitwise operator [ << , >> , | , & , ~ , ^ ].

Assignment operator [ += , -= , \*= , /= , %= ].

# 04 Data Type in C

◇ Data type in C

## Data type in C

Int (it has at least 2 bytes , usually 4 bytes in size and its format is %d , %i ).

Char (it has 1 bytes in size and its format is %c ) .

Float (it has 4 bytes in size and its formate is %f ).

Double (it has 8 bytes in size and its format is %lf).

Note : for compile > alt + f9

for run > ctrl + f9

for output show > atl + f5



# 05 condition statements in C

- ◇ Simple if
- ◇ If else
- ◇ If else if
- ◇ Nested if

## Simple if

```
if(condition){  
    //statements  
}
```

## if else

```
if(condition){  
    //statements  
}else{  
    //statements  
}
```

## nested if

```
if(condition){  
    if(condition){  
        //statements  
    }else{  
        //statements  
    }  
}else{  
    //statements  
}
```

## If else if

```
if(condition){  
    //statements  
}else if{  
    //statements  
}else{  
    //statements  
}
```

# 06 Loop in C

- ◊ While loop
- ◊ Do while loop
- ◊ For loop

## While loop

### Syntax

#### Initialization

```
While(condition){  
    //statement  
    inc/dec;  
}
```

## for loop

### syntax

```
for(initialization ; condition ; inc/dec){  
    //statement  
}
```

## do-while loop

### syntax

```
initialization;  
do{  
    //statement  
    inc/dec  
}
```

# 07 Switch

## ◇ Switch

### Switch

```
initialization  
switch(condition){  
    case : //statement;  
        break;  
  
    default : //statement;  
}
```



# 08 Array

## ◆ What is array

### What is array

arrays are used to store multiple values in a single variable , instead of declaring separate variables for each value.

example :

```
#include<stdio.h>
void main(){
    int num[5];
    int i;
    printf("enter the number : ");
    for(i = 0 ; i < 5 ; ++i){
        scanf("%d" , &num[i]);
    }
}
```

# 09 Function

- ◈ User defined
- ◈ Pre - defined

## User defined

no return no argument  
with return no argument  
no return with argument  
with return with argument

## pre – defined

printf();  
scanf();  
getch();  
clrscr();  
main();

# 10 pass by value & pass by reference

- ◇ Pass by value
- ◇ Pass by reference

## Pass by value

Pass by value means that you pass actual value of the variable into the function .

Example :

```
Int sum (int o , int z);
Void main(){
    int result = sum(5 , 6);
    printf("%d",result);
}

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

## Pass by value

Pass by value means to pass the reference of an argument in the calling function to the corresponding formal parameter of the called function .

Example :

```
Int sum (int *no1 , int *no2);
Void main(){
    int x =10;
    int y = 30;
    int result = sum (&x , &y);
    printf("result is : %d",result);
    printf("\n x is : %d",x);
    printf("\n y is : %d",y);
}

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

# 11 Union & structure

- ◊ Union
- ◊ structure

## Union

union in C is a special data type available in c that allows storing different data types in the same memory location .

```
example : union[union name ]
        {
            member definition ;
            member definition ;
            .....
            member definition ;
        };
```

## structure

structure is a keyword that creates user – defined data type in c/c++. A structure creates a data type that can be used to group items of possibly different type into a single type .

```
example : struct[structure name ]
        {
            member definition ;
            member definition ;
            .....
            member definition ;
        };
```