C Instruction

Types of instruction :

There are three types of instruction

- The Declaration :-

This instruction is used to declare the type of variables being used in the program. The type declaration statement is written at the beginning of main() funcation.

Ex: - int bas;

float rs, grossasal;

char , name ,code;

- **Arithmetic Instruction: -** A C - arithmetic consists of a variable name on the left hand side of = and variable name and constants on the right hand side of= the variables and constants appearing on the right hand side of = are connected by arithmetic operators(+,-,/).

ex :- int ad;

float kot, deta, alpha,beta, gamma;

ad= 3200; kot=0.0056; deta= alpha*beta/gamma+3.2*2/5;

- **Control Instruction :-** This instruction are use to control the sequence of execution of berries statement in a C.

(a) Sequence control instruction

(b) Selection of decision control instruction

(c)Repetition or loop control instruction

(d) Case control instruction

The types of Conversion in Assignments:-

The types of expression on right hand side and the type of the variable on the left hand side of an assignment operator may be not same.

the value of the expression is promoted or demoted depending on the type of the variable on left hand side of=

Ex:- int I;

float b;

i=3.5; b= 30;

Hierarchy of Operation :-

- While executing an arithmetic statement that has multiple operators.
- There might be issues about their evaluation.

Ex :- 2*X-3*Y or (2x)-(3y) or 2(x-3y)?

Associatively of Operators :-

Ex:- a=3/2*8; a=b=3;

The priority or precedence in which the operations in an arithmetic statement are performed is called the hierarchy of operations.

- **Example 1.1:** Determine the hierarchy of operations and evaluate the following expression: -
- i = 2 * 3 / 4 + 4 / 4 + 8 - 2 + 5 / 8
- Stepwise evaluation of this expression is shown below:

i = 2 * 3 / 4 + 4 / 4 + 8 - 2 + 5 / 8

i = 6 / 4 + 4 / 4 + 8 - 2 + 5 / 8

i = 1 + 4 / 4 + 8 - 2 + 5 / 8

i = 1 + 1+ 8 - 2 + 5 / 8

i = 1 + 1 + 8 - 2 + 0

i = 2 + 8 - 2 + 0

i = 10 - 2 + 0

i = 8 + 0 i = 8

Note that 6 / 4 gives 1 and not 1.5. This so happens because 6 and 4 both are integers and therefore would evaluate to only an integer constant. Similarly 5 / 8 evaluates to zero, since 5 and 8 are integer constants and hence must return an integer value.

Example 1.2: Determine the hierarchy of operations and evaluate the following expression:

kk = 3 / 2 * 4 + 3 / 8 + 3

Stepwise evaluation of this expression is shown below:

kk = 3 / 2 * 4 + 3 / 8 + 3

kk = 1 * 4 + 3 / 8 + 3

kk = 4 + 3 / 8 + 3

kk = 4 + 0 + 3

kk = 4 + 3

kk = 7

Note that 3 / 8 gives zero, again for the same reason mentioned in the previous example. All operators in C are ranked according to their precedence. And mind you there are as many as 45 odd operators in C, and these can affect the evaluation of an expression in subtle and unexpected ways if we aren't careful.

Associativity of Operators

- Associativity can be of two types—Left to Right or Right to Left. Left to Right associativity means that the left operand must be unambiguous.
- Consider the expression a = 3 / 2 * 5 ;

Here there is a tie between operators of same priority, that is between / and *. This tie is settled using the associativity of / and *. But both enjoy Left to Right associativity.

Operator	Left	Right	Remark
/	3	2 or 2 * 5	Left operand is unambiguous, Right is not
*	3 / 2 or 2	5	Right operand is unambiguous, Left is not

Since both / and * have L to R associativity and only / has unambiguous left operand (necessary condition for L to R associativity) it is performed earlier.

Consider one more expression a = b = 3 ;

Here both assignment operators have the same priority and same associativity (Right to Left).

Operator	Left	Right	Remark
=	a	b or b = 3	Left operand is unambiguous, Right is not
=	b or a = b	3	Right operand is unambiguous, Left is not

Since both = have R to L associativity and only the second = has unambiguous right operand (necessary condition for R to L associativity) the second = is performed earlier.

Control Instructions in C

- (a) Sequence Control Instruction
- (b) Selection or Decision Control Instruction
- (c) Repetition or Loop Control Instruction
- (d) Case Control Instruction

QUESTIONS:-

Convert the following equations into corresponding statements.

(a) $Z = \dfrac{8.8(a+b)2/c-0.5+2a/(q+r)}{(a+b)*(1/m)}$

(b) $X = \dfrac{-b+(b*b)+2\ 4ac}{2a}$

2. Evaluate the following expressions and show their hierarchy.

(a) g = big / 2 + big * 4 / big - big + abc / 3 ;

(abc = 2.5, big = 2, assume **g** to be a float)

(b) on = ink * act / 2 + 3 / 2 * act + 2 + tig ;

(ink = 4, act = 1, tig = 3.2, assume **on** to be an int)

What would be the output of the following programs:

(a) main( )

```
{
int i = 2,
j = 3,  k, l ;
float a, b ;
k = i / j * j ;
l = j / i * i ;
a = i / j * j ;
b = j / i * i ;
printf( "%d %d %f %f", k, l, a, b ) ;
}
```

(b)
```
main( )
{
   int a, b ;
   a = -3 - - 3 ;
   b = -3 - - ( - 3 ) ;
   printf ( "a = %d b = %d", a, b ) ;
}
```

(c)
```
main( )
{
   float a = 5,  b = 2 ;
   int c ;
   c = a % b ;
   printf ( "%d", c ) ;
}
```

(d)
```
main( )
{
   printf ( "nn \n\n nn\n" ) ;
   printf ( "nn /n/n nn/n" ) ;
}
```

(e)
```
main( )
{
   int a, b ;
   printf ( "Enter values of a and b" ) ;
   scanf ( " %d %d ", &a, &b ) ;
   printf ( "a  = %d b = %d", a, b ) ;
}
```