C PROGRAMMING HANDBOOK BY SAICHANDAN GORLI

CHAPTER 2 : INSTRUCTIONS AND OPERATIONS

A C program is a set of instructions. Just like a recipe which contains instructions to prepare a particular dish

TYPES OF INSTRUCTIONS

1. Type declaration Intructions.

2. Arithmetic Instrictions.

3. Control Instructions.

TYPE DECLARATION INSTRUCTIONS

This is how you declare a variable in C.

```
int a;

float b;

char c;
```

OTHER VARIATIONS:

Some other variations of this declaration look like this:

```
int a;  // Declare an integer variable 'a'
float b;  // Declare a float variable 'b'
int i = 10; #  // Declare and initialize 'i' with 10
int j = i; // # Declare 'j' and initialize with 'i'
int a = 2, b = 3, c = 4, d = 5;  // Declare and initialize multiple variables

int j1 = a + j - i; //Valid: use previouly defined varibles

 // Invalid : 'a' is used before declaration
 // float b = a + 3;
 // float a = 1.1;

 // Valid: Assigning the same value to multiple variables

int a,b,c,d;
a = b = c = d = 30;
```

ARITHMETIC INSTRUCTIONS

Arithmetic instructions perform mathematical operations.

Here are some of the sommoly used program in C language:

- +(Addition)
- -(Subtraction)
- /(Division)
- *(Multiplication)
- %(Modulus)

Note:

1. Operands can be int/float etc. **+-*/** are arithmetic operators.

```
int a = 2;
int b = 4;
int z =  a * b; // legal
int z;
a * b = z; // illegal
```

2. % is the modular divisor operator
    - % → returns the remainder
    - % → cannot be applied on float
    - % → sign is same as of numerator (14 % 2 == 0)
3. No operator is assumed to be present.

```
int i = ab; // invalid
int i a * b; // valid
```

4. There is no operator to perform exponentiation in C hwoever we can use pow (x,y) from <math.h>.

TYPE CONVERSION

An Arithmetic operation between

- int and int → int
- int and float → float
- float and float → float

Example:

```
#include <stdio.h>
int main()
{

    int a = 9;
    float b = 2;
    float c = a / b;
    printf("the value of a / b is : %f", c);
    return 0;
}
```

- 9 / 2 becomes 4 as both the operand are int
- 9 / 2.0 becomes 4.5 as one of the operands is float
- 2 / 9 becomes 0 sa both the operand are int

NOTE:

In programming, type compatibility is crucial. For int ***a = 2.5;*** =, the float 2.5 is demoted to 3, losing the fractional part because ***a*** is an integer. Conversely, for ***float a = 8;***, the integer 8 is demoted to 8.0, matching the float type of ***a*** and retaining precison.

```
int a = 2.5; // In this case 2.5 (float) wil be demoted to 2 (int) because a is not able to store floats.

float b = 8; // b will be store 8.0 | 8 => 8.0
```

OPERATOR PRECEDENCE IN C

Have a look a below statement:

3*x-8*y is (3x)-(8y) or 3(x-8y)?

In C language simple mathematical rules like BODMAS, no longer apply.

The answer to above questions is provided by operatoe precedence & associativity.

OPERATOR PRECEDENCE

The following table lists the operator priority in C

Priority	**Operators**
1st	* / %
2nd	+ -
3rd	=

Operators of higher priority are evaluated first in the absence of parenthesis.

OPERATOR ASSOCIATIVITY

When operators of equal priority are present is an expression, the tie is taken care of by associativity.

x * y / z → (x * y) / z

x / y * z → (x / y) * z

CONTROL INSTRUCTIONS

Determine the flow of the in a program four types of control instructions in C are :

1. Sequence Control Instructions.
2. Decision Control Instructions.
3. Loop Control Instructions.
4. Case Control Instructions.