Chapter 2: Instructions and Operators

A C program is a set of Instructions. Just like a recepie – which contains Instructions to prepare a particular Dish.

Types of Instructions

1> Type declaration Instruction
2> Arithmetic Instruction
3> Control Instruction

Type declaration Instruction
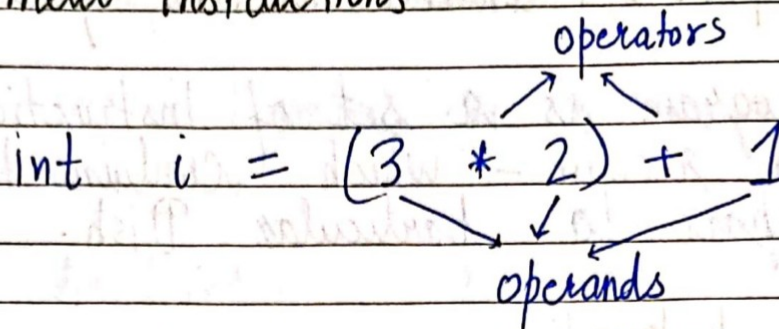
```
int a;
float b;
```

Other Variations:

```
int i=10; int j=i; int a=2
int j1= a + j - i;
```

float $b = a+3$; float $a = 1.1$ => ERROR! as we are trying to use a before defining it.

```
int a, b, c, d;
a = b = c = d = 30;
```
⟹ Value of a, b, c & d will be 30 each.

Arithmetic Instructions

operators

$$\text{int } i = (3 * 2) + 1$$

operands

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

```
int b = 2, c = 3;
int z; z = b * c;      ✓ legal
int z;  b * c = z;     ✗ Illegal (Not allowed)
```

% → Modular division operator
% → Returns the remainder
% → Cannot be applied on float
% → Sign is same as of numerator ($-5 \% 2 = -1$)

$5 \% 2 = 1$ $-5 \% 2 = -1$

Note :-

1, No operator is assumed to be present

```
int i = a b      → Invalid
int i = a * b    → Valid
```

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

Type Conversion

An Arithmetic operation between

Int and Int → Int
Int and float → Float
Float and float → Float

$5/2 → 2$ $5.0/2 → 2.5$ } Important!!

$2/5 → 0$ $2.0/5 → 0.4$

Note :-

int a = 3.5;   In this case 3.5 (float) will be
demoted to 3 (int) because a is
not able to store floats.

float a = 8;   a will store 8.0
       $8 → 8.0$ (promotion to float)

Quick Quiz:
Q.   int k = 3.0/9   Value of k? and why?

S. $3.0/9 = 0.333..$ but since k is an int,
it cannot store floats & value
0.33 is demoted to 0.

Operator precedence In C

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

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

The answer to the above question is provided by operator 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 in an expression, the tie is taken care of by associativity.

$$x * y / z \implies (x * y) / z$$

$$x / y * z \implies (x / y) * z$$

$*$, / follows Left to right associativity

Control Instructions
Determines the flow of Control in a program
Four types of Control Instructions in C are:

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