Instructions

These are statements in a Program.

Types

- Type Declaration Instructions
- Control Arithmetic Instructions
- Arithmetic Instructions

Type Declaration Instructions -> Declare var before using it

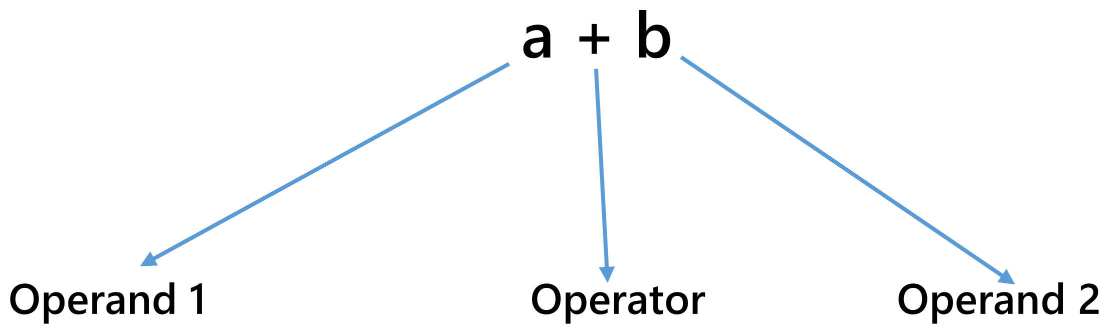
VALID

```
int a = 22;
int b = a;
int c = b + 1;
int d = 1, e;
int a,b,c; a = b = c = 1;
```

INVALID

```
int a = 22;
int b = a;
int c = b + 2;
int d = 2, e;
int a,b,c = 1;
```

Arithmetic Instructions



NOTE - single variable on the LHS

VALID

a = b + c
a = b * c
a = b / c

INVALID

b + c = a
a = bc
a = b^c

NOTE - pow(x,y) for x to the power y

Type Conversion

int op int -> int

int op float -> float

float op float -> float

Operator Precedence – BODMAS(Bracket Opening, Division, Multiplication, Addition, Subtraction)

Associativity (for same precedence)

Left to Right -> x = 4 * 3 / 6 * 2

Control Instructions

Used to determine flow of program

- Sequence Control
- Decision Control

- Loop Control
- Case Control

Operators

- Arithmetic Operators
- Relational Operators
- Comparison Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Ternary Operator

Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	x++
--	Decrement	Decreases the value of a variable by 1	x--

Assignment Operators

Assignment operators are used to assign values to variables.

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<==	x <<= 3	x = x << 3

Comparison Operators

Comparison operators are used to compare two values (or variables). This is important in programming, because it helps us to find answers and make decisions.

The return value of a comparison is either 1 or 0, which means true (1) or false (0). These values are known as Boolean values

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Logical Operators

You can also test for true or false values with logical operators.

Operator	Name	Description	Example
&&	Logical AND	Returns true if both statements are true	x < 5 && x < 10
||	Logical OR	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical NOT	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

Bitwise Operators in C

- **The & (bitwise AND) - takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1.**

- **The | (bitwise OR) - takes two numbers as operands and does OR on every bit of two numbers. The result of OR is 1 if any of the two bits is 1.**

- **The ^ (bitwise XOR) - takes two numbers as operands and does XOR on every bit of two numbers. The result of XOR is 1 if the two bits are different.**

- **The ~ (bitwise NOT) - takes one number and inverts all bits of it.**

X	**Y**	**X & Y**	**X | Y**	**X ^ Y**
0	**0**	**0**	**0**	**0**
0	**1**	**0**	**1**	**1**
1	**0**	**0**	**1**	**1**
1	**1**	**1**	**1**	**0**

Ternary Operator

It can be used to replace multiple lines of code with a single line. It is often used to replace simple if else statements.

variable = (*condition*) ? *expressionTrue* : *expressionFalse*;