

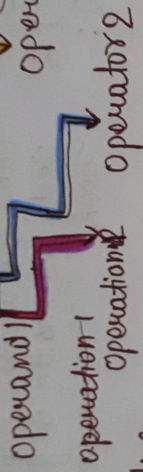
Lesson-

Expression & Conversions

• What is C Expression

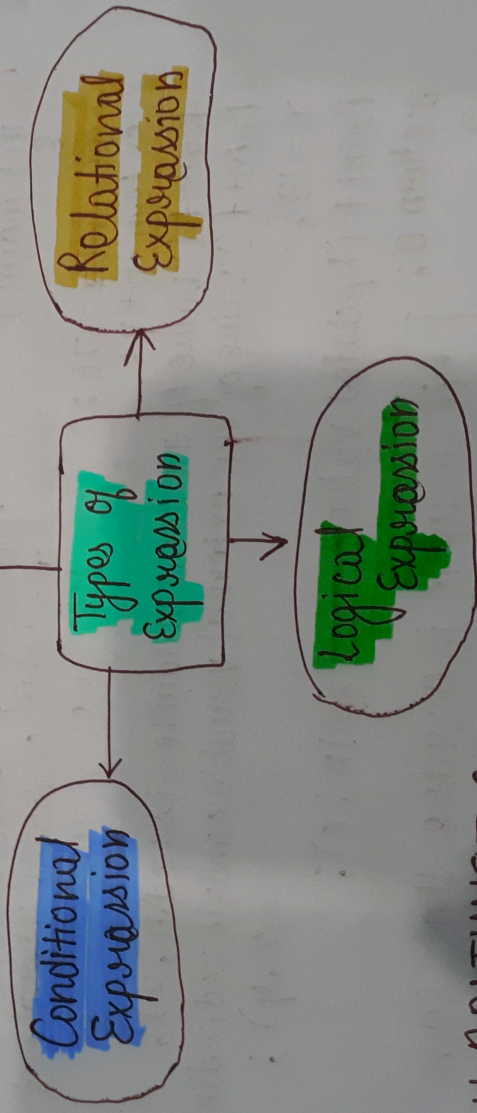
Q- What is an expression?

result =  $a + b * c \dots$



As a combination of operands and operators → It computes a single value stored in a variable. The operator denotes the action or operation to be performed. The operands are the items to which we apply the operation.

• Types of expressions



ARITHMETIC EXPRESSION

$A + B * C$

$6 / 4 * 2$

→ An arithmetic expression is an expression using addition (+), subtraction(-), multiplication (\*), division(/), and exponentiation (\*\*). A single arithmetic expression is an expression all of whose operands are of the same type. (i.e. integer, Real, or complex)

2) Relational Expression → An arithmetic expression is an expression used to compare two operands. It is a condition which should be taken or not.

- $X \neq 3 == 0$
- $A > B$
- $8 < 3$

3) Logical Expression → A logical expression is a statement that evaluates to either "true" or "false". Logical expressions are considered false when equal to 0, and are considered true when nonzero.

- $(x < 10) \&\& (y > 5)$
- $7 > 4 \parallel 6 < 7$
- $A + b < C \&\& !D$

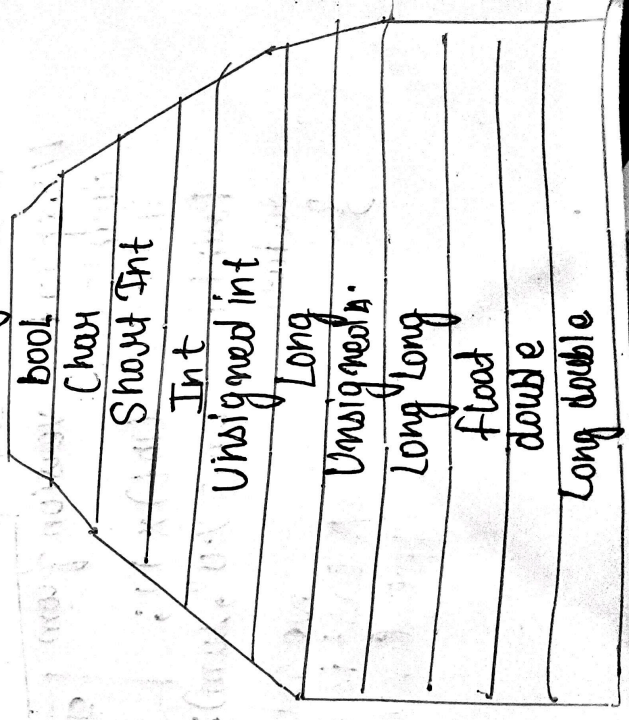
4) Conditional Expression → The conditional expression is the decision making statements which depends upon the output of the expression.

```

Status = (Age > 22) ? 'H' : 'U';
EVEN = (N % 2 == 0) ? 1 : 10;
    
```

• Types of Conversion

1. Implicit Conversion
2. Explicit Conversion, Implicit Type Conversion



# 1. Implicit Conversion → Implicit Expansion Conversion.

// An example of implicit conversion

```
#include <stdio.h>
int main()
{
    int x = 10; // integer x
    char y = 'a'; // character c
    // y implicitly converted to int. ASCII
    // value of 'a' is 97
    x = x + y;
    // x is implicitly converted to float
    float z = x + 1.0;
    printf("x = %d, z = %f", x, z);
    return 0;
}
```

An Implicit Conversion Sequence is the sequence of conversions required to convert an argument in a function call to the type of the corresponding parameter in a function declaration.

# 2. Explicit Conversion

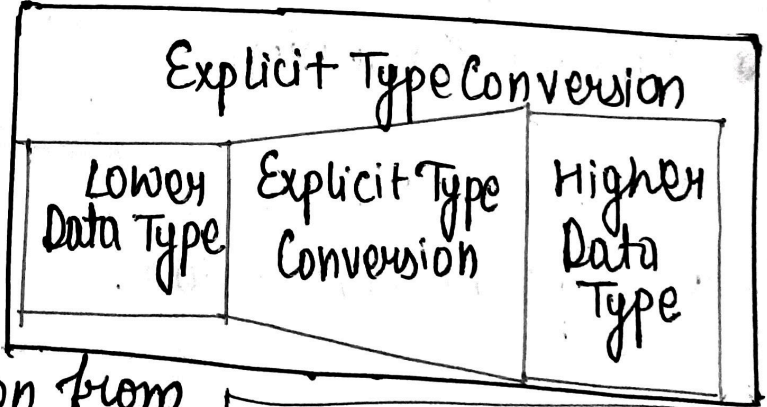
#include <stdio.h>

int main()

```
{
    double x = 1.2;
```

// Explicit Conversion from double to int

```
int sum = (int) x + 1;
printf("sum = %d", sum);
return 0;
}
```



~~Imp~~ Explicit Conversion: To create a conversion between two incompatible types,

we must use a casting. A cast is simply an explicit type conversion.