Declaring a Variable:

A variable declaration specifies the type and name of the variable:

(int , float , double , char)

int age: Declaring an integer variable named 'age'

float pi: Declaring a floating-point variable named 'pi'

char grade: Declaring a character variable named 'grade'

Initializing a Variable:

Initialization means assigning an initial value at the time of declaration:

int age = 20: Integer variable with an initial value of 25

float pi = 3.14: Float variable with an initial value of 3.14

char grade = 'A': Character variable initialized with 'A'

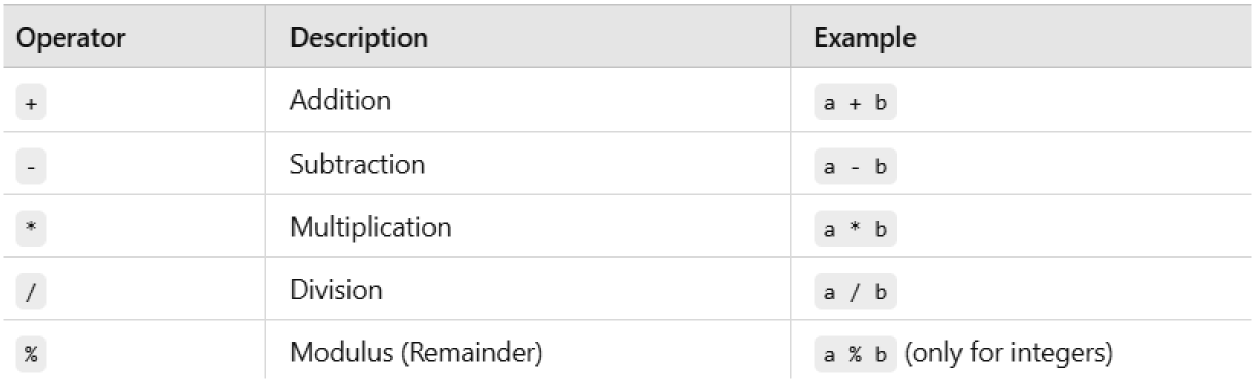
Performing Arithmetic Operations :

Integer Operations: +, -, *, /, and % are used with integer variables.

Floating-Point Operations: +, -, *, / work with float and double, but % is not allowed for floating-point numbers.

%.2f Format Specifier: Used to print floating-point numbers with 2 decimal places.

%%: Used to print % in printf.



Operator	Description	Example
`+`	Addition	`a + b`
`-`	Subtraction	`a - b`
`*`	Multiplication	`a * b`
`/`	Division	`a / b`
`%`	Modulus (Remainder)	`a % b` (only for integers)

Types of Conditional Statements

1. if Statement:

   Executes a block of code only if a condition is true.

2. if-else Statement:

   Executes one block if the condition is true, otherwise executes another block

3. if-else if-else Statement:

   Used when you have multiple conditions to check.

4. switch Statement:

   Used when there are multiple possible values for a variable.

5. Ternary Operator (? :) (Shorthand for if-else):

6. A compact version of if-else.

Loops in C:

Loops allow executing a block of code multiple times based on a condition.

Types of Loops in C:

1. **for loop** – When you know how many times to repeat.

   ```
     for(initialization; condition; update) {
           // Code to execute
     }
   ```

2. **while loop** – When you repeat while a condition is true.

   Used when the number of iterations is **unknown** (runs while the condition is true).

   ```
   while(condition) {
      // Code to execute
   }
   ```

3. **do-while loop** – Like while, but always runs **at least once**.

   ```
   do {    // Code to execute
        } while(condition);
   ```

Using Constants in C:

Constants are fixed values that do **not change** during the execution of a program. In C, you can define constants using:

1. const Keyword:

- Prevents accidental modification of values.
- Improves code readability.

2. #define Preprocessor Directive:

   - No memory allocation (used during pre-processing).
   - Cannot be changed in the program.

3. enum for Named Constants:

   - Makes code more readable.
   - Useful when dealing with related constants.

Age in days problem:

https://codeforces.com/group/MWSDmqGsZm/contest/219158/submission/311918570

last 2 digits:

https://codeforces.com/group/MWSDmqGsZm/contest/219158/submission/311925287

Sequence of Numbers and Sum:

https://codeforces.com/group/MWSDmqGsZm/contest/219432/submission/311931371

Divisors:

https://codeforces.com/group/MWSDmqGsZm/contest/219432/submission/312111966