C Programming

Variables and Constants

A variable is essentially a container where we can store different pieces of information.(Numbers, texts, characters…)

- int = The most natural size of integer for the machine
- float = A single-precision floating point value.
- double = A double-precision floating point value.
- char = Typically a single octet(one byte). It is an integer type.
- void = Represents the absence of type.

Format Specifiers

- %c = Character
- %d = Signed Integer
- %f = Floating Point
- %s = String

- %g or %G = Similar as %e or %E
- %hi = Signed Integer(short)
- %hu = Unsigned Integer(short)
- %i = Signed Integer
- %l or %ld or %li = Signed Integer
- %lf = Floating Point
- %Lf = Floating Point
- %lu = Unsigned Integer
- %lli or %lld = Signed Integer
- %llu = Unsigned Integer
- %o = Octal representation of Integer
- %p = Address of pointer to void void
- %e or %E = Scientific Notation of Float Values
- %u = Unsigned Integer
- %x or %X = Hexadecimal representation of Unsigned Integer
- %n = Prints Nothing
- %% = Prints % Character

Working with Numbers

Any operation you do between a floating point number and integer is going to return a floating point number back

Any operation you do between two integers will give an integer back

- ceil = Returns the nearest integer which is more than or equal to the argument passed to this function
- floor = Returns the nearest integer which is less than or equal to the argument passed to this function

- round = Returns the nearest integer value of the float/double/long double argument passed to this function(.1 to .5= <   .6 to .9= >)
- sin = Sinus
- cos = Cosinus
- tan = Tangent
- sinh = Hyperbolic Sinus
- cosh = Hyperbolic Cosinus
- tanh = Hyperbolic Tangent
- exp = Exponential
- log = This function is used to calculate natural logarithm
- log10 = This function is used to calculate base 10 logarithm
- sqrt = This function is used to find square root of the argument passed to this function
- pow = This function is used to find the power of the given number
- trunk = This function truncates the decimal value from floating point value and returns integer value

Comments

Comment is a special block of code in C which actually gets ignored when we run our program

Constants

Constant is a special type of variable in C which can't be modified.

Arrays

An array is essentially a data structure where we can store a bunch of different data values

Array indexes starts at 0

Functions

A function is basically a collection of code that performs a specific task

Operators

Arithmetic Operators

- + = Adds two operands.
- - = Subtracts second operand from the first.
- * = Multiplies both operands.
- / = Divides numerator by de-numerator.
- ++ = Increment operator increases the integer value by one.
- — = Decrement operator decreases the integer value by one.

Relational Operators

- == = Checks if the values of two operands are equal or not. If yes, then the condition becomes true.
- != = Checks if the values of two operands are equal or not. If the values are not equal, then the condition becomes true.
- > = Checks if the value of left operand is greater than the value of right operand. If yes, then the condition becomes true.
- < = Checks if the value of left operand is less than the value of right operand. If yes, then the condition becomes true.
- >= = Checks if the value of left operand is greater than or equal to the value of right operand. If yes, then the condition becomes true.
- <= = Checks if the value of left operand is less than or equal to the value of right operand. If yes, then the condition becomes true.

Logical Operators

- && = Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.
- || = Called Logical OR Operator. If any of the two operands is non-zero, then the condition becomes true.
- ! = Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.

Structures

A struct is a data structure where we can store groups of data types

While Loops

A While loop is basically a structure in C programming language that we can use to loop over and continually execute a specific block of code until a certain condition is false

For Loops

A For loop is a special type of loop that we can use in C which allows us to use something called an "indexing variable".

An indexing variable will basically tell us what iteration of the loop we're certainly on.

Two Dimensional Arrays

A two dimentional array is basically a situation where we have an array where all of the elements in that array is actually arrays themselves.

Nested For Loop

A nested for loop is a situation where we have a for loop and inside of that for loop we have another loop.

Pointers

A pointer is just a type of data that we can use inside of our programs.

File Modes

A file can be opened in different modes. Below are some of the most commonly used modes for opening or creating a file.

- r : opens a text file in reading mode.
- w : opens or creates a text file in writing mode.
- a : opens a text file in append mode.
- r+ : opens a text file in both reading and writing mode. The file must exist.
- w+ : opens a text file in both reading and writing mode. If the file exists, it's truncated first before overwriting. Any old data will be lost. If the file doesn't exist, a new file will be created.
- a+ : opens a text file in both reading and appending mode. New data is appended at the end of the file and does not overwrite the existing content.
- rb : opens a binary file in reading mode.
- wb : opens or creates a binary file in writing mode.
- ab : opens a binary file in append mode.
- rb+ : opens a binary file in both reading and writing mode, and the original content is overwritten if the file exists.

- wb+: opens a binary file in both reading and writing mode and works similar to the w+ mode for binary files. The file content is deleted first and then new content is added.
- ab+: opens a binary file in both reading and appending mode and appends data at the end of the file without overwriting the existing content.