

Fundamentals of C Programming Notes with LaTeX2e and KOMA-Script

A. Data Types

→ (signed) char: 1 byte min: -128, max: 127 stores numbers in 2's complement notation
→ int: 2/4 bytes, min (2 byte): -32768, max (2 byte): 32767, min (4 byte): -2147483648, max (4 byte): 2147483647
→ float: 4 bytes, min: -1.2e-38 (+-), max: 3.4e+38 (+-)
→ double 8 bytes, min: 2.3e-308 (+-), max: 1.7+308 (+-) float/double use IEEE floating point notation (sign/exponent/mantissa).

B. Operators

→ - subtraction (addition of 2's complement)
→ % remainder/modulus
Order of precedence: execute left to right Process only two values at a time. The tree structure is called the evaluation tree.
High
() - Paranthesis
+ - () (unary ie sign of number 1, -1)
* /+ - (binary ie in equation)
=
Low

C. Type Casting

int a = 3, int b = 2; double c;
for integer division of $c = a/b$ $c = 3/2 = 1$;
for double division: $c = 3.0/2.0 = 1.5$;
if integer division is not the preferred operation use type cast
 $c = (\text{double}) a / b$; OR $c = a / (\text{double}) b$; OR
 $c = (\text{double}) a / (\text{double}) b$;

D. Operators

→ - subtraction (addition of 2's complement)
→ % remainder/modulus
Order of precedence: execute left to right Process only two values at a time. The tree structure is called the evaluation tree.
High
() - Paranthesis
+ - () (unary ie sign of number 1, -1)
* /+ - (binary ie in equation)
=
Low

E. Proceudral Programming

C is a procedure (function) oriented programming language.
Every C program has at least one function – main().
Functions allow us to logically group statements that performs a specific task.
Breaking down a program to sensible number of functions is important.
> Human readability of the code
> Code reuse
> Exhaustive testing
Local variables of a function are created in the beginning of function execution and destroyed after. Function has multiple inputs (copies of local variables) but either has 1 value to output or none.

F. Control Structures

A Control structure is an element that influence the current point of execution
> Sequence
> Selection (sometimes called 'alternation')
> Repetition (sometimes called 'iteration')
Control structures execute compound statements - Blocks.
...
C variables have block scope – a variable can only be accessed inside the block it was defined in.

G. Conditional Operators

Conditional operators are involved in the comparison of two values (evaluation tree).
The only issue is, be careful when comparing float (or double) types with int types
2.3 is stored as 2.2999999523162841796875 in float, so there is a loss of accuracy in conversion or comparison.
In C logical TRUE is represented with 1, logical FALSE with 0

H. For loop

initialisation, condition and variation Step 1: initialisation Step 2: condition Step 3: Body of loop Step 4: variation Step 2,3,4 are then repeated.

I. Order of Logical Operations

High
Function calls
! + - (unary)
* / %
+ -
< <= >= >
== !=
||
=
Low
Short circuit evaluation when the condition can be evaluated before reaching the end of the expression ie if(costlyfunction1() || costlyfunction2() || costlyfunction3())
where costlyfunction1() is evaluated as true so the rest are ignored.

J. Arrays and strings

Arrays - allocate block of contiguous memory to store a fixed number of data elements of the same type
Array size is always passed as an int.
A Function has no direct way to find out the size of the array.
> A local copy is not made, changes to the array in the function are reflected in the original Array.
> If an array element is passed to a function, a local copy is made.
> const keyword when passing an array to a function that should remain unchanged.
> Strings can be defined in two ways:
char str[/*size*/]; - allows changing the string. char* str; - cannot change the string after it has been assigned once.

K. Structs

Structs save a local copy when passed.
struct patient{
int id;
char first_name[MAX_NAME_LENGTH+1];
char second_name[MAX_NAME_LENGTH+1];
int date_of_birth;
int phone_number;
};
typedef struct patient patient_node_t;
struct patient_list{
struct patient_list_t *next;
patient_node_t patient;
};
typedef struct patient_list patient_list_t;

L. Abstraction

Procedural abstraction: separation of what a function does from the details of how the function accomplishes its purpose.
eg. (String str="Hello world" String s1=str.substring(0,6)) I know what substring does, it returns a substring of the characters from 0-6 but I don't know how it does it.
Data Abstraction: separation of the logical view of a data object (what is stored) from the physical view (how the information is stored). (e.g. use struct to store custom defined data types). People don't know what the data does.
Information Hiding: protecting the implementation details of a lower-level module from direct access by a higher-level module. basically a combination of the benefits of procedural and data abstraction.