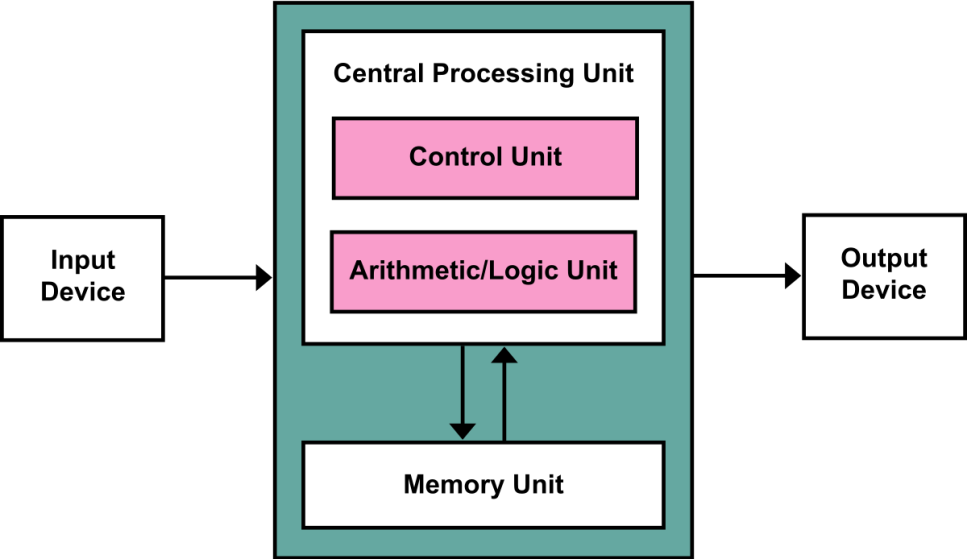
**What is a C program?**

* It’s Just a Text File
* Understandable format language
* What we refer to as High-Level language OR Human readable language

**von Neumann diagram**



Not accurate computer diagram, but is a good enough model for computers

1. Memory
2. Control Unit
3. Arithmetic Logic Unit
4. Input / Output

**Memory**

* **Store data** and **instructions** for program execution

1. **What is Instructions**
   1. Instructions are basic expression of the basic operations
   2. E.g. + , - , \*, /
   3. “ { } “ -> body of a function
   4. “ ( ) “ -> parenthesis parameter
   5. E.g.
      1. Operation = ‘+’
      2. Source operands = x, y
      3. Destination operands = z
   6. Op-code -> representation of operator in bits
      1. Example “+” is represent as 01100 in bits
2. **RAM**
   1. Basic cycle of CPU
      1. Fetch -> Decode -> Execute -> Write Back -> Fetch
      2. Fetch
         1. Fetch from memory to control unit
      3. Decode
         1. Breaking down the information to op-code and operands
      4. Execute
         1. Processed the information in the ALU for result
      5. Write Back
         1. Write Back the result to memory and return the cycle
3. **What is ASCII?**
   * Idea to represent each character with 8 bit (0-255)
   * 0-127 (Readable Character)
   * 128-255 (Unreadable Character)

**Directives**

* Lines in a C program that begins with a ‘#’ symbol
  + #include <stdio.h>
  + <stdio.h> is to be found in the library-include in the user directories
  + “stdio.h” is in the user-defined directories, default is in the same directory as the C file
* During Pre-processing
  + Read libraries and remove comments

**Variable**

* Objects that represent real world thing/idea (simulation / representation)
* Reality : computers only read one or zero
* Assign values to variable, so that the value mean something in the program
* If a program is successfully compiled and linked, each DECLARED variable is assigned a specific memory location during Runtime

**“string literally” is whatever in between the “” with an un-escaped character**

**Identifies**

* Name
* Variables and function have names
* Only these alphabets can be used
  + Digits (0-9)
  + Letters (a-z, A-Z)
  + Underscore ( \_ )

**A statement is a Line ends with ;**

e.g. compiled statement:

int main(){  
}

if (expression){  
}

**else** is paired with the closest **un-paired if**

if() //#1  
 if() //#2

else{} //paired with #2

Syntax:

int I\_AM\_a\_VAR;

int \_IAMOK\_2;

int \_1234\_OK;

int I-am-Error; //compile\_error, ‘-‘ is an operator

int 2\_AsaC\_ERROR; //compile\_error, ‘2’ is a number at the front

Declaration/un-Initialize:

int a;

float b;

Initialization:

int a = 10;

float b = 10.0f;

Assignment/Assignment:

a = 20;

b = 20.0f;

-------------------------------------------------------------------------

Declaration + Definition

in .h file:

extern int a; // Declaration

extern static float b;

struct A

{

static int iampublicdefault; // Users have access

private:

static int iamprivate; // Users have no access

};

class B

{

static int iamprivatedefault; // Users have no access

public:

static int iampublic; // Users have access

};

in .cpp file

int a = 10; // Definition

static float b = 10.0f;

static int A::iampublicdefault = 10;

static int A::iamprivate = 10;

static int B::iamprivatedefault = 10;

static int B::iampublic = 10;