Learn Programming Basics (C Language)

LESSON #001
INTRODUCTION to Programming

Basic programming using Programming C

1. introduction
2. some Unix/Linux command using cygwin
3. stages of Compiling
4. introduction, data types
5. printf, scanf, Precedence Table
6. if, else if, else, ternary, switch, break
7. for loop, while, do while, continue, break
8. function declaration, definition nation, read complex declaration
9. array, read complex declaration
10. pointer, pointer and array relation, double pointer
11. malloc(memory allocation)
12. struct, union
13. struct memory layout, size of struct/union

never move onto the next topic if absolute zero idea on how pointer works.

C++ topic

1. namespace, scoping
2. r reference
3. CPP class, default ctor, ctor, copy ctor, const function, ctor/dtor order
4. CPP compiler generate ctor, class member initialization, this,
5. Operator function overload
6. vector, string, stack, queue etc... copy ctor methods
7. function template
8. function overloading
9. read complex declaration 2
10. iterator
11. class template, template specialization
12. template T
13. inheritance

Advanced C++

1. move ctor, constructor/dtor order recap
2. virtual function, virtual table, abstract class
3. inheritance with virtual function
4. read complex declaration 3, a little bit on goto:,
5. static_cast, dynamic_cast, reinterpret_cast, const_cast
6. ref collapsing rules
7. functor, lamda
8. std::forward, std::move,
9. varadic template
.. type erasure, meta programming, tuple, reflection etc…
10. On your own, good luck

Why learn C and C++ for starters?

- By learning to drive an auto car, requires you to relearn manual car.
- By learning to drive a manual car, you can learn auto car on you own.
- C and C++ is just like a manual car.
- The rest of any programming languages is an auto car.

What is typing?

- The act of typing text using a keyboard.
- The act of typing text on a text-based program in a human-readable languages such as English etc…

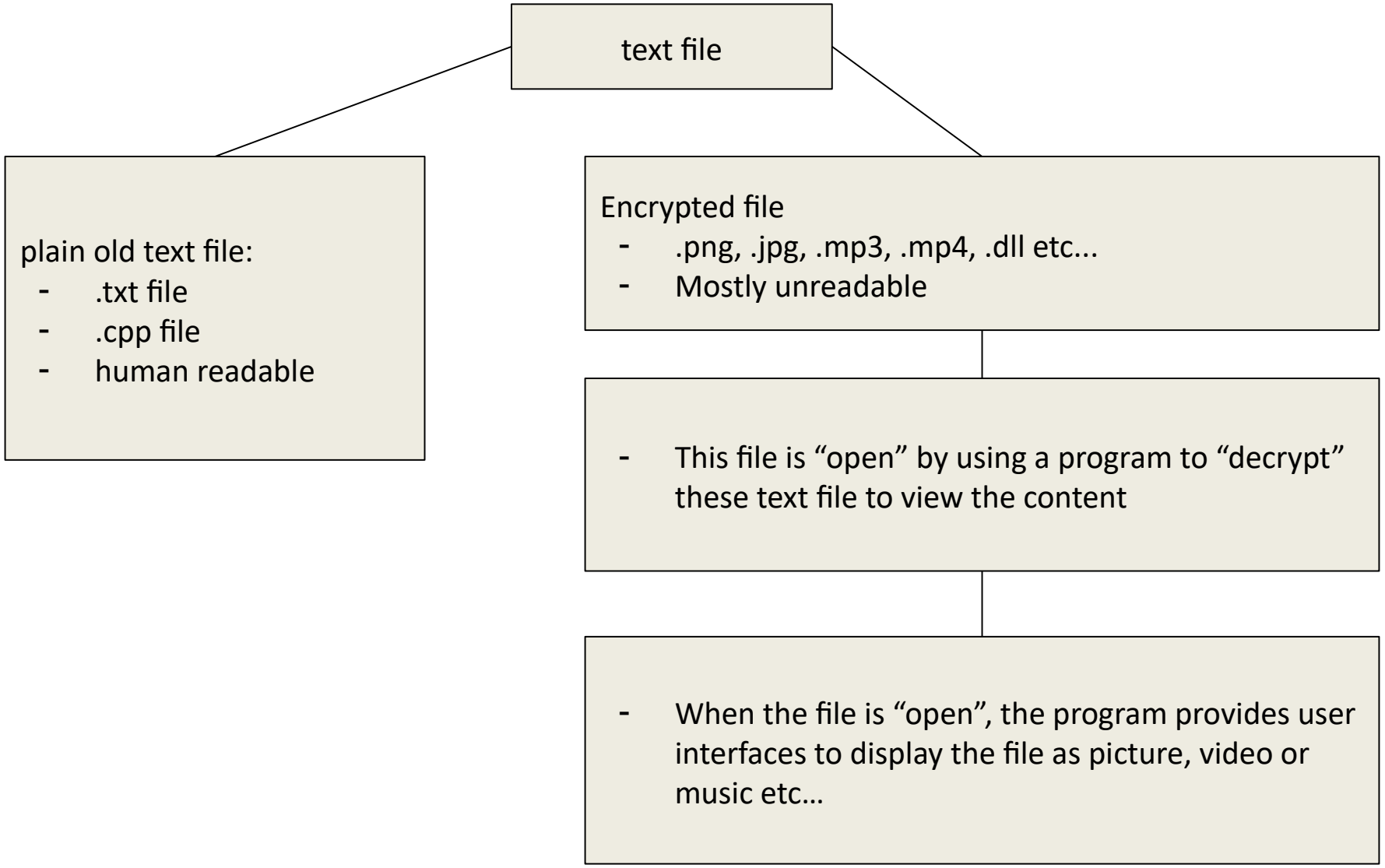
What is programming?

- The act of typing text using a programming language.

2 type of things on a computer

- There are only 2 main categories in a computer
  - text file (.txt, .png, .jpg, .mp3, .mp4 etc…)
  - application (.exe file, cmd console, game.exe, etc…)

What is a text file? (details)



What is a text file? (details)

- It is a pretty gray area on the what's define a text file, but this is just a very simplified meaning

- Technically there is really only 2 main categories in a computer
  - text file (.txt, .png, .jpg, .mp3, .exe etc…)
  - application (exe file, binary file)

  Folders is a just a feature to categories things in different location

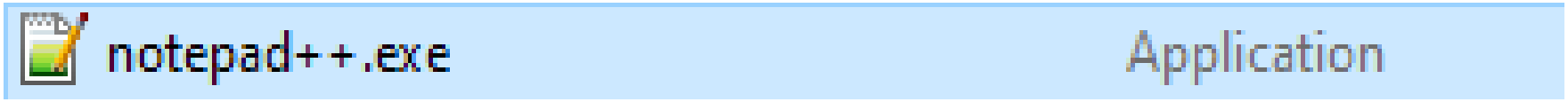
What is a computer program?

- a computer program is an executable file (.exe)
- It's also known as application

What is a computer program? (Details)

- the compilation of 1 or more text file(s).
- the compilation of all the necessary text files needed to create the computer program.

- After a successful compile, the computer-readable binary file is generated known as executable file (.exe) in windows OS.
- Example: notepad is a computer program

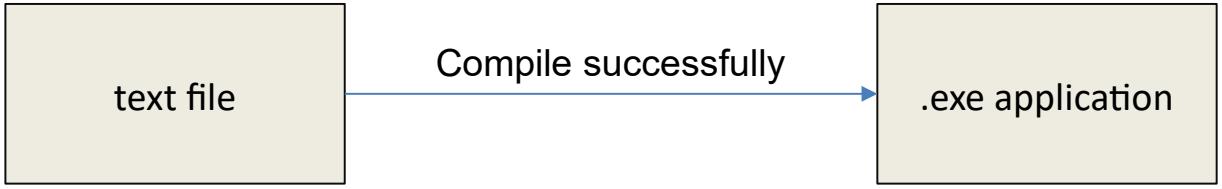


How does a program being created? (Details)

- An executable file (.exe) is generated by compiling text file(s) using its' respective compiler.

- A compiler is a program that converts a text file into instructions and then into machine-code, in 1 and 0 known as binaries, so that the program can be executed or "read" by a computer.

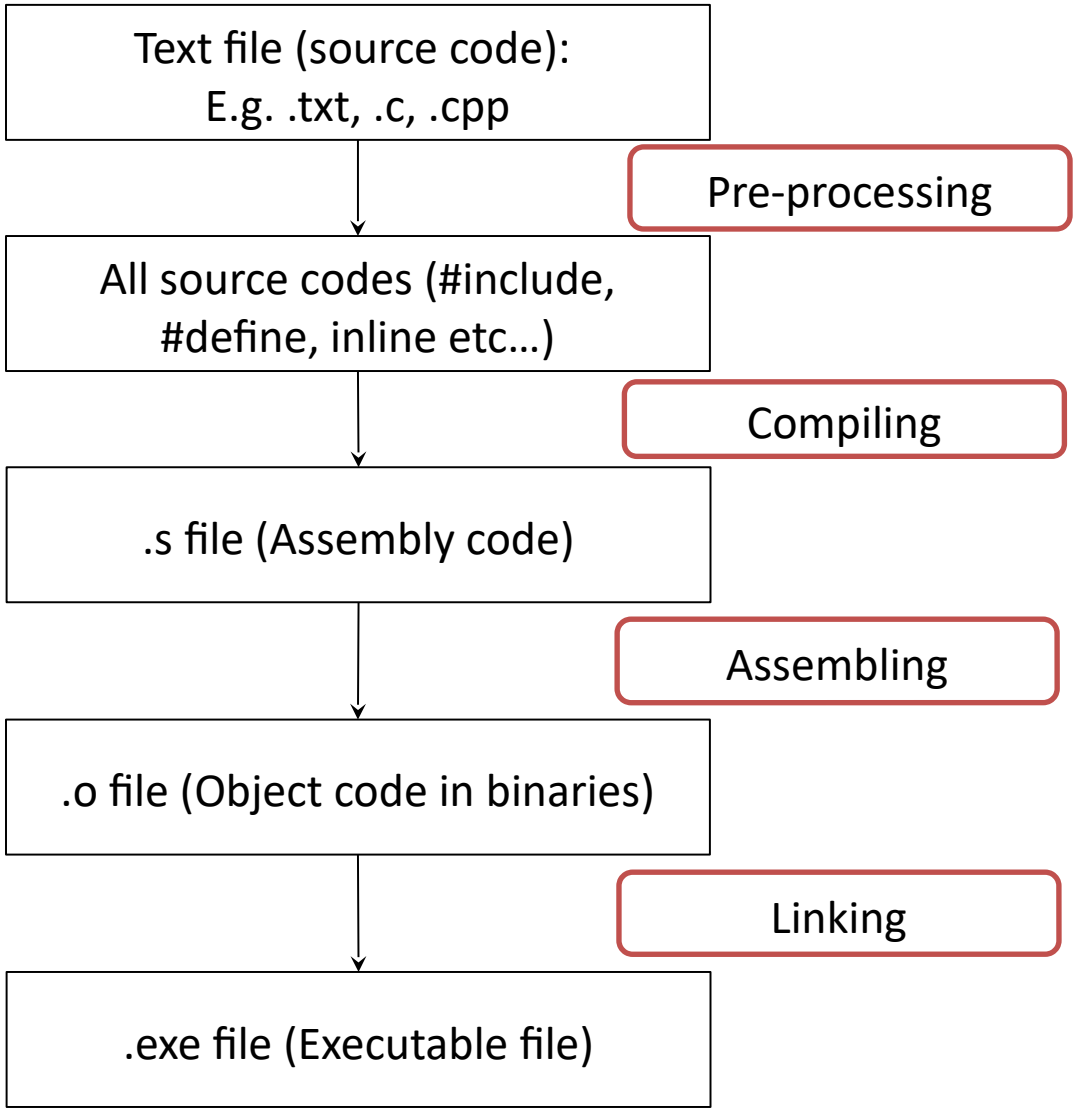
- Therefore, a computer program is referred to or being called as an **executable file OR an application OR .exe file**



How to run a computer program?

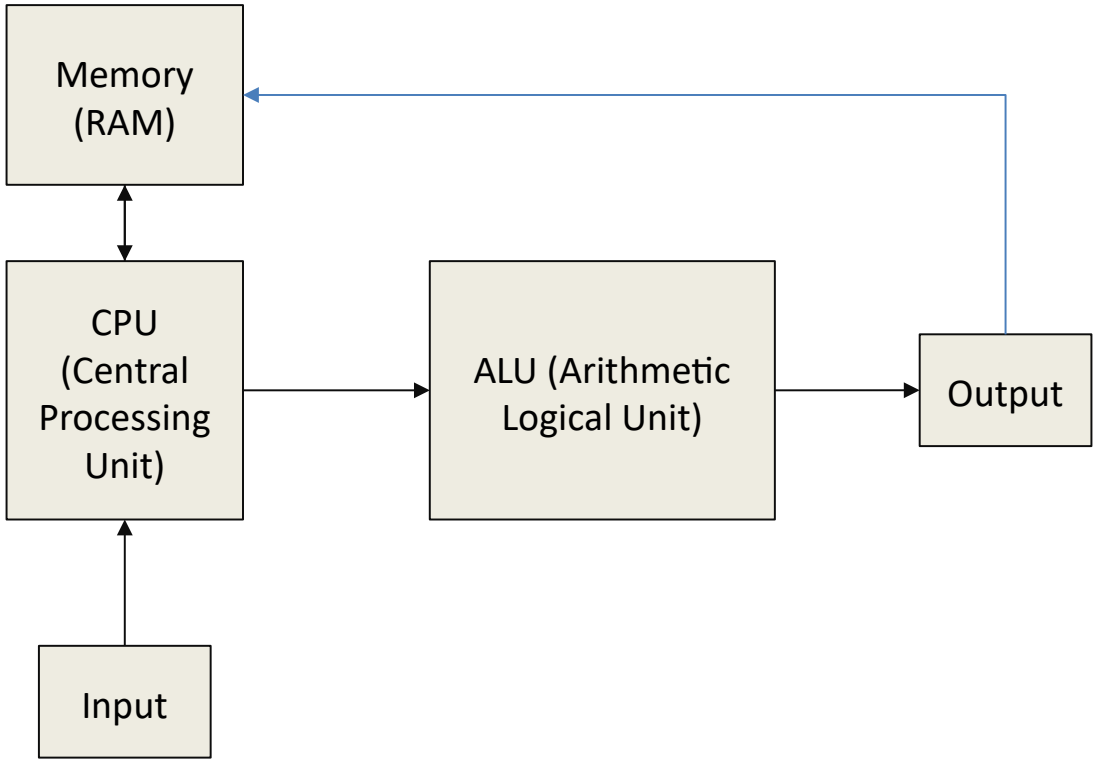
- **double clicking** the (.exe) file (automatically, it's a feature)
- select the (.exe) file and press enter
- run using command console (manually)
- etc…

Stages of compiling overview



Von Neumann Architecture

- A super simplified diagram of a computer diagram



4 Basic cycle of computer system (CPU)

- **Fetch**
  - Fetch data from memory/control unit.
- **Decode**
  - breaks down the information to opcode and operands.
- **Execute** (ALU)
  - processed the decoded data in the ALU for results.
- **Write Back**
  - write back the result to storage.
  - or use it as output.
  - or return cycle etc…
- Predict? Revert? …

Memory - RAM (Random Access Memory)

- Stored data and instruction for program execution temporary.
- Temporary allocated space for computer programs to run it's program instructions.

What is instruction?

- Instruction are expression of basic operation.
- Computer instruction are made up of op-code, operands and sources.
- Basic expressions of basic operations:
  - e.g. +, -, *, /

Example: Opcode

- z <- x + y
  - x, y, z are what we called operands
    - x, y are called sources.
    - z are called destination operands.

- Representation of instructions in 1 and 0

+	z	x	y

- Encoded

0x5B	0x9B	0x7B	0x6B

Etc…

History

Leibniz and the I Ching [edit]


Gottfried Leibniz

Leibniz studied binary numbering in 1679; his work appears in his article *Explication de l'Arithmétique Binaire* (published in 1703). The full title of Leibniz's article is translated into English as the *"Explanation of Binary Arithmetic, which uses only the characters 1 and 0, with some remarks on its usefulness, and on the light it throws on the ancient Chinese figures of Fu Xi"*.[19] Leibniz's system uses 0 and 1, like the modern binary numeral system. An example of Leibniz's binary numeral system is as follows:[19]

- 0 0 0 1 numerical value 2^0
- 0 0 1 0 numerical value 2^1
- 0 1 0 0 numerical value 2^2
- 1 0 0 0 numerical value 2^3

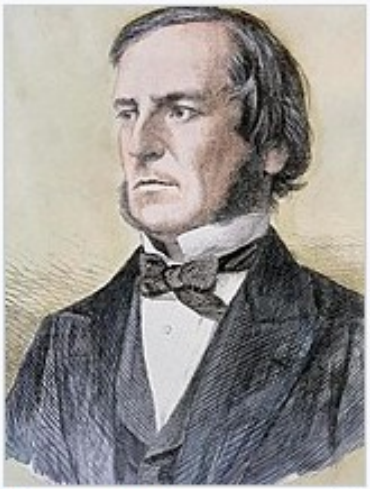
Leibniz interpreted the hexagrams of the I Ching as evidence of binary calculus.[20] As a Sinophile, Leibniz was aware of the I Ching, noted with fascination how its hexagrams correspond to the binary numbers from 0 to 111111, and concluded that this mapping was evidence of major Chinese accomplishments in the sort of philosophical mathematics he admired. The relation was a central idea to his universal concept of a language or characteristica universalis, a popular idea that would be followed closely by his successors such as Gottlob Frege and George Boole in forming modern symbolic logic.[21] Leibniz was first introduced to the *I Ching* through his contact with the French Jesuit Joachim Bouvet, who visited China in 1685 as a missionary. Leibniz saw the *I Ching* hexagrams as an affirmation of the universality of his own religious beliefs as a Christian.[20] Binary numerals were central to Leibniz's theology. He believed that binary numbers were symbolic of the Christian idea of *creatio ex nihilo* or creation out of nothing.[22]

> [A concept that] is not easy to impart to the pagans, is the creation *ex nihilo* through God's almighty power. Now one can say that nothing in the world can better present and demonstrate this power than the origin of numbers, as it is presented here through the simple and unadorned presentation of One and Zero or Nothing.
>
> — Leibniz's letter to the Duke of Brunswick attached with the *I Ching* hexagrams[20]

History

- What if I can calculate math automatically?

Later developments [edit]



George Boole

In 1854, British mathematician George Boole published a landmark paper detailing an algebraic system of logic that would become known as Boolean algebra. His logical calculus was to become instrumental in the design of digital electronic circuitry.[23]

In 1937, Claude Shannon produced his master's thesis at MIT that implemented Boolean algebra and binary arithmetic using electronic relays and switches for the first time in history. Entitled *A Symbolic Analysis of Relay and Switching Circuits*, Shannon's thesis essentially founded practical digital circuit design.[24]

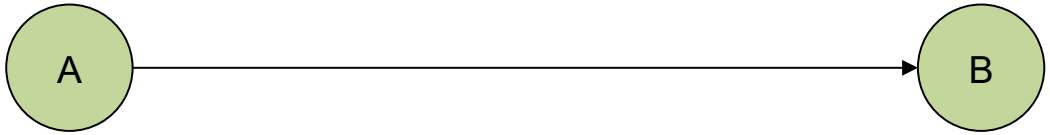
In November 1937, George Stibitz, then working at Bell Labs, completed a relay-based computer he dubbed the "Model K" (for "**K**itchen", where he had assembled it), which calculated using binary addition.[25] Bell Labs authorized a full research program in late 1938 with Stibitz at the helm. Their Complex Number Computer, completed 8 January 1940, was able to calculate complex numbers. In a demonstration to the American Mathematical Society conference at Dartmouth College on 11 September 1940, Stibitz was able to send the Complex Number Calculator remote commands over telephone lines by a teletype. It was the first computing machine ever used remotely over a phone line. Some participants of the conference who witnessed the demonstration were John von Neumann, John Mauchly and Norbert Wiener, who wrote about it in his memoirs.[26][27][28]

The Z1 computer, which was designed and built by Konrad Zuse between 1935 and 1938, used Boolean logic and binary floating point numbers.[29]

What is an algorithm?

- A very detailed step by step instructions to solve problems.
- It can be written in point form but mostly written in the form of flowchart for better visual representation.

Example how do you pick a box from point A to point B?



1. Start at point A in front of the box
2. Bend left elbow 90 degree anti-clockwise
3. Bend right elbow 90 degree anti-clockwise
4. Bend your back 90 degree clockwise
5. Shift Left palm towards right palm
6. Can left palm continue shifting towards right? Yes - to step 7, No to step 5
7. Shift right palm towards left palm
8. Can right palm continue shifting towards left ? Yes - to step 9, No to step 7
9. Bend your back 90 degree anti-clockwise
10. Now walk algorithm
11. blah blah blah bend thigh, bend knee, bend toes, Ski-bi dibby dib yo da dub dub
12. reach point B? No to step 11 yes to step 13
13. Stop

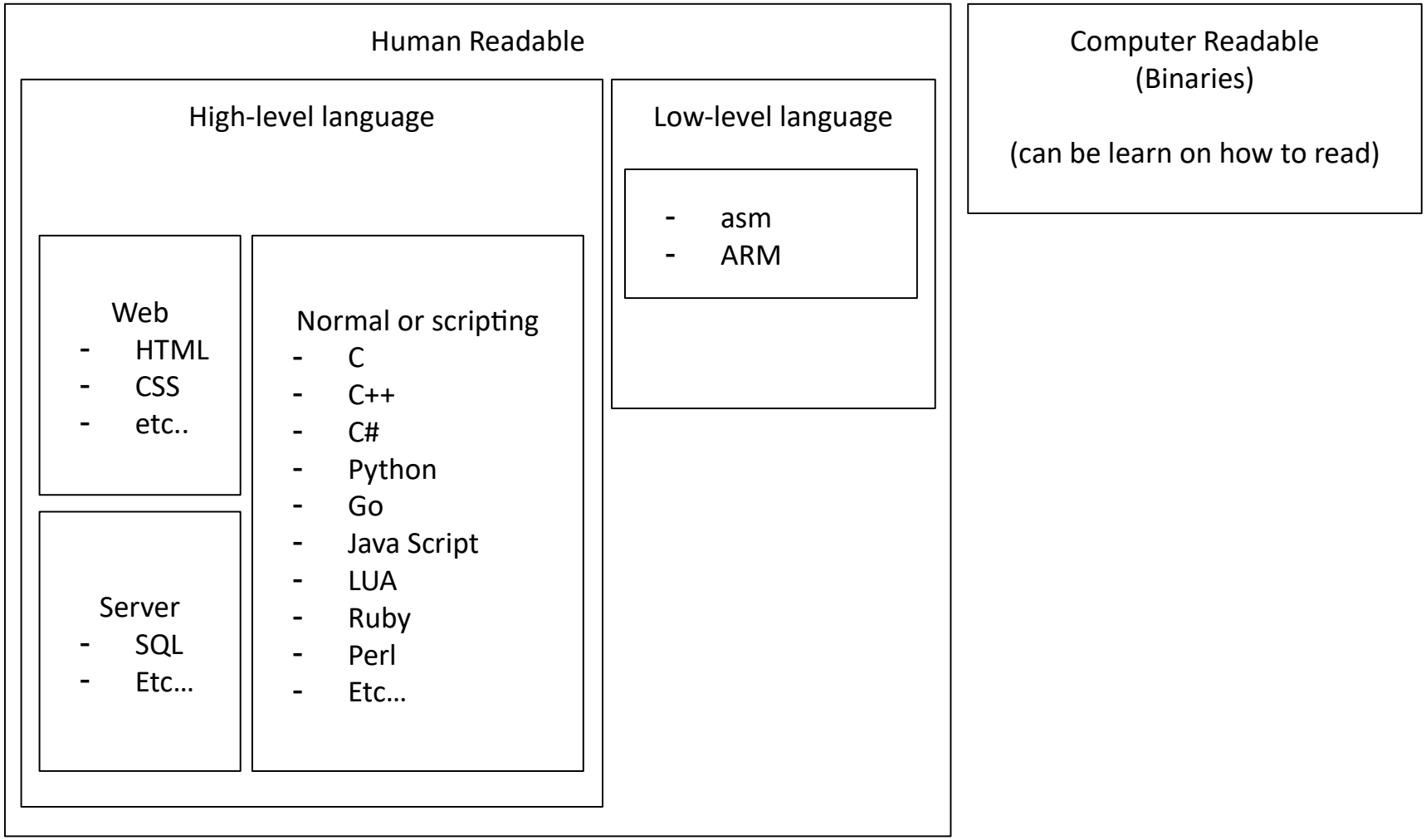
GCD Euclid Algorithm (300 B.C)
find the Greatest Common Divisor

1. Start
2. Let a and b be 2 positive number
3. if a < b, go to step 4, else go to step 5
4. let r = remainder of b/a (b mod a), go to step 6
5. let r = remainder of a/b (a mod b)
6. let if r = 0 then go to step 8
7. let a = b, b = r, go to step 5
8. GCD is b
9. stop

Example:

1. a = 24, b = 18
2. 24 < 18
3. r = 6 (24 mod 18)
4. a = 18, b = 6
5. r = 0 (18 mod 6)
6. GCD = 6
7. stop

Type of programming language



A basic program memory layout