6th Semester Engineering :→

C, Python

History of Programming Languages :→

1942 → Denis Ritchie + Team

(A)

↳ 1838 → Ada Lovelace

↳ UNIX → μc & μP

A + mod^n → B

1972 → C →

ANSI

→ American National Standards Institute

JIT → AICTE, NAAC, NBA → BE / B TECH

1st Std Version → 1989

C + SIMULA → C++ → Denmark, Bjarne Stroustrup

Dijkstra's

C + C++ → Java → 1995, James Gosling → Bean

Lab → Sun Micro Systems ↳ Oracle

1991 → Guido Van Rossum → Monty [Python] Flying Circus

Pointers		
C	Python	RUST
C++	Java	
Block 1	Block 2	

```c
#include <stdio.h>   // → Header File, Preprocessor Directive, Standard I/O Library
int main () {
printf ("JIT");
return 0;
}
```

How does a C program get executed?

notepad / VS code (Source code) → .c → Compile (gnu mingw) → 1010 / 0101 object → MLL Binary → .exe → lib → executable → runner → O/P

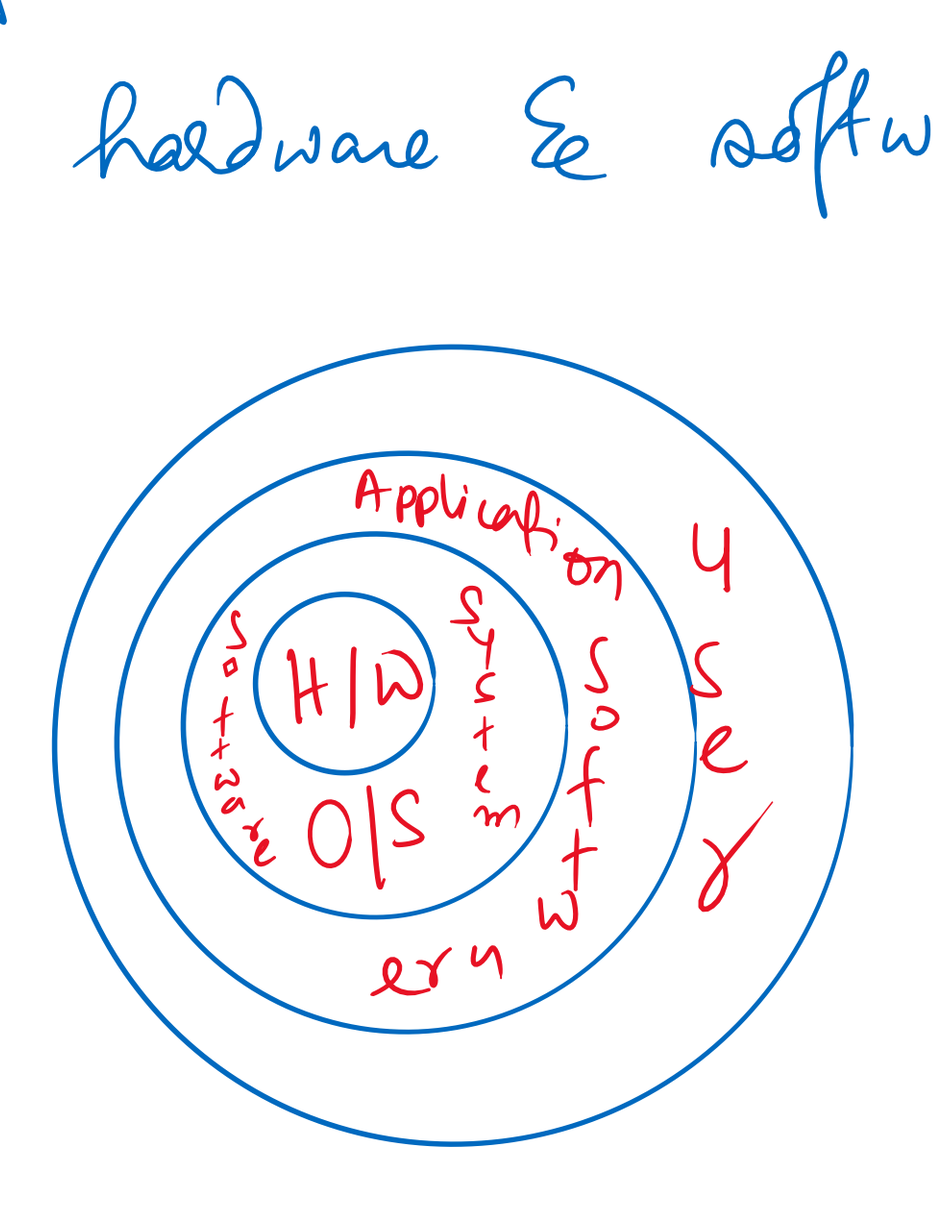
Software ⟷ Hardware

Software → CS | IS | AIML | AIDS

Hardware → ECE, EEE, MECH

- System Software
  - Operating System — Win, Mac, Linux, Ubuntu
  - System Support — Driver, Update
  - System Development — Editors
- Application Software
  - General Purpose — MS Office
  - Application Specific — WhatsApp

Relation b/w hardware & software :→



Data Types in C :→ Categories

Primary	void	Enumerated	Derived
Integer: short, int, long	↓ void	↓ enum	Arrays
Floating: float, double		Named Constants	Strings
Character: char		Mon - Sun	Pointers
		Jan - Dec	Struct
		→ Int → 4 bytes	Union
			C++ class

Short → −128 to 127 → 2 bytes

int → -2^{31} to $2^{31}-1$ → 4 bytes

(-2^{31} ↓ INT_MIN, $2^{31}-1$ ↓ INT_MAX) → C++

Java → Integer.MaxValue or MinValue

long → Some Coding Questions. → 8 bytes

float → 4 bytes

double → 8 bytes

CLION ↓ C & C++

```c
int a = 10;
int b = 20;
{ a + b = 30; }
```

→ Expression (a, b → operand; + → operator)

a, b are also called variables or identifiers.

(int)(int) = 10; ✗

Naming Conventions :→

1a ✗ a1 ✓

a b ✗ 1ax ✗

$a ✓

_a ✓ → Interface

functions :→

Camel case: rateOfInterest();

Pascal case: Classes RateOfInterest();

	Format Specifiers	C++
integer →	%d or %i	No format specifiers
float →	%f or %0.1f, 0.2f	
double →	%lf or %0.x lf	
char →	%c	
sizeof() →	%llu (long long unsigned) or %zu (binary unsigned)	
String →	%s	

* Operators :→

(I) Arithmetic :→ +, −, /, %, *

Python: **, // (exp, fd)

(II) Assignment Operators :→

↳ Augmented ✓ =, +=, −=, /=, %=, *=, **=, //=

↳ Shorthand ✓ a = a+2, a (+=) 2

(III) Logical → &&, ||, ! → short cut and, or, not

(IV) Comparison / Relational :→ >, <, >=, <=, ==, != ↳ Boolean

(V) Unary Operators :→ Inc / Dec ++, −− (Prefix, Postfix)

(VI) Conditional / Ternary Operator :→

C/C++ (condition) ? trueValue : falseValue;

data type match

Java :→

DataType val = (condition) ? tv : fv;

* Bitwise Operators :→ Bit Masking / Manipulation

6 operators: Int — Int → (Int) (bin, bin)

1. Bitwise AND → &
2. " OR → |
3. " XOR → ^
4. " Right Shift → >>
5. " Left Shift → <<
6. " NOT → ~

5 → 0101
7 → 0111
5 & 7 → 0101 = 5

6 → 0110
8 → 1000
6 | 8 → 1110 = (14)

XOR → 10 → 1010, 11 → 1011 → 0001

S → 0, D → 1

Shift Operations :→

value, a = 10

unit / step = 2

a >> 2 ↓

8 bit version

| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ~~1 0~~ → discarded

10 >> 2 = 1 (should be 2)

11 >> 2 → 2

11 << 3 →

a << 2

10 << 2 = 40

~~0 0~~ discarded | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | (32, 8) → 40

| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ~~1 1~~ = 2

~~0 0 0~~ | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | (64 32 16 8 4 2 1)

64 + 16 + 8 = (88)

Bitwise Not Operator :→

~5 = −6 (PV)

~5 = 10

NOT GATE TRUTH TABLE

I/P	O/P
T	F
F	0

I/P	O/P
1	0
0	1

5 → 0101

~5 → 1010 = 10. Theoretical Value

(−6) = 10 ✓

−6 = abs (−6)

= 6 → 0110 → 1's complement

1001

0001 +1

1010 → 10

2's complement (1's c + 1)

MCQs → ~5 → 10, 1, 2, (−6) → PV

Formula: $n \rightarrow \sim n = -n-1$

$40 = -41$

$-500 = -(-500) - 1 = 499$