

C - Programming language

- "#" in header file is an pre-processor of an header file.
- GCC = GNU Compiler Collection. → Compiler in C - lab.
- C - language is an programming language made by Dennis Ritchie @ 1970.
- Computer system is an Electronic device that stores, manipulates and retrieves data.
- Hardware is an device which we can touch or feel whereas Software is a Set of Inst., data which is used To operate Computers and Execute Specific Tasks.
- 5 Components Model of Computer
 - Input (Giving Input To memory)
 - Output
 - Memory (Primary, Secondary, Cache)
 - CPU (ALU + CU + Registers)
 - Storage devices (Operating system, Magnetic systems)

* CPU = Arithmetic logic unit (ALU) + Control Unit (CU) + Registers ...

→ Programming language is an Intermediate language which is used To communicate b/w low level language and High level language (English) Using device.
(Binary) HLL \rightleftharpoons Programming language \rightleftharpoons LLL

Note → C - language is an High Level language.

⇒ Compilation process in an Computer

- Files are Source File , header file, object file and Executable file.
- Source file Contains The Source Code of The program whereas Header file Contains The sub- routines in diff. program which The user wants To use. object file are generated by The Compiler as a result of processing The source code file.
- Pre- processor (#) is used for compilation and it is process of Converting the source code into The object code and it is done by Compiler. Compiler Checks The source code and Syntax Error, If Source Code is Error free, Then, Compiler will Generate object code. "#" Takes The source code as Input into The object code or The machine code. "#" removes comments from code.
- Software Refers To The set of Electronic program Instructions /data, a Computer processes goods in order To perform Tasks / operations.
- 1 Byte Contains 8 Bit whereas 1 Nibble Contains 4 Bit.

Good Write

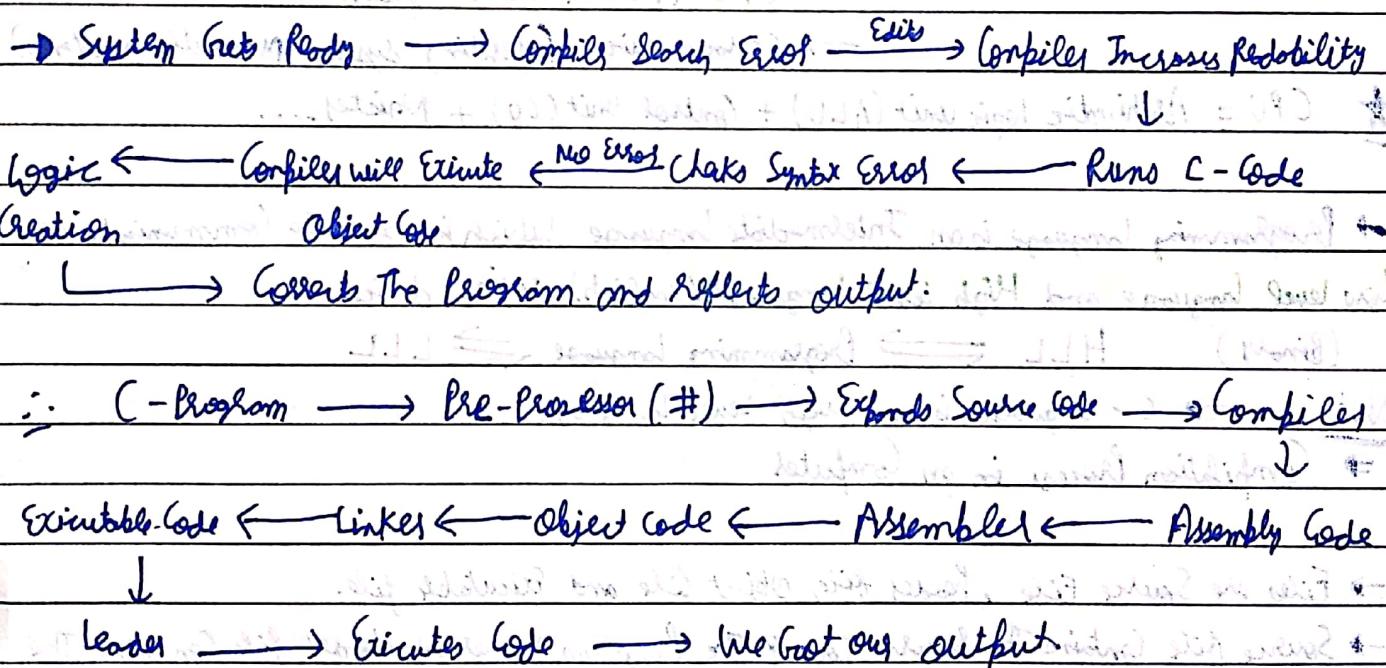
System Software (SS)

Application Software (AS)

- SS is used for operating Computer hardware.
- SS are Installed on The Computer when operating System Is Installed.
- In Actual, User is not Interacting with SS as it is Running in Background.
- SS Can Run Independently. It Provides Platform for Running.
- ① Eg:- Windows

- AS is used to Perform Specific Task.
- AS are Installed Acc. To User Requirements.
- User is Interacting with AS.
- AS Can't Run Independently. They Can't Run Without The Help of SS.
- ① Eg:- Microsoft Word

⇒ Compilation Process In detail



⇒ Compiling Process

- The Code is The Source Code stored on the desk must be Translated Into an machine language and This is The Job of The Compiler. The Compiler is an Computer device That Translates The Source Code Written in High level language and This is Called as Compilation Process.
- The C-Compiler is having 2 Separate Programs → Pre-Processor → Translates in Input Language
- The Pre-Processor Reads The Source code and prepares it for Translation.

Good Write

→ While Preprocessing The Code, it looks for Special Instructions Known as Preprocessor Commands (Eg: - "#include <conio.h>") and These Commands are from Special Code Libraries.

→ After The Pre-Processor has Prepared The Code for Compilation, The Compiler does The Actual Work of Converting The Program into Machine Code.

- ⇒ Computing Environments → Personal Computing Environment. ①
→ Client Server. ②
→ Distributed Computing Environment. ③
→ Grid-Sharing Computing Environment. ④

- ① = Use of Personal Computer
② = Working in Non-MNC
③ = In Distributed Computing, all The Machines, Clients, Servers Share The Processing Tasks.
④ = The Concept of Grid-Sharing Computing is To Share the Processing of The Computer Basic on The Criteria of Time. In This Type of Env., The Complete Processing is done by Central Computer.

⇒ Algorithm, flow chart and Pseudo Code

1. Algorithm Gives The Logic of Program i.e. Step by step description of How To Achieve a given Soln. It must be written in such a way that can be easily converted to Computer Instructions.

② Characteristic of Algorithm → Finiteness, Unambiguous, Input, Output, Feasible.

2. Flowchart are Representation of a Program by Collection of Diagrammatic Representations and Symbols Connected with Arrows and representing seq. of Instructions.

3. Pseudo Code is Compact & Informal Code. It is Human-Readable and it is consist of short English Phrases / Paragraphs are used Here. It is Consist of Both Pros and Cons of a Project and In any Company, Pseudo Code is only limited to Top HR & CEO.

⇒ Tokens (with their symbol & their definition) are
- Identifier, Constant, Operator, etc.

→ These are The Basic Building Blocks of 'C' language.

→ Each and Every Smallest Unit in Programme is called as 'C' Token.

Good Write

* Keywords are reserved words in C-language.

→ Identifiers

→ This is The General Terminologies for The names of Variable, rⁿ & arrays.

→ Rules for Naming Identifiers is :-

- ① It must begin with a letter or underscore.
- ② It mustn't Contain any Spaces.
- ③ It must Consist of only letters, digits or underscore.
- ④ Special Characters are Not allowed in Name of Variable.

→ A Variable is Nothing but an name given To an Storage Area that Used to memmory.

→ Rule for Creation of Variable is :-

- ① The Name of The Variable is Composed of letters, digits and underscore.
- ② It must begin with either a letter or underscore.

* Places we can declare a Variable is :-

① When a Variable is declared inside a function, then it is called local variable.

② When a Variable is declared outside a function, then it is called 'global variable'.

→ When a Variable is declared in the ~~func~~ Block of a function, Then, it is called as formal parameters. But The Variables which are provided during the functional call are actual parameters.

→ Constants in C-language

→ A C-Const. refers To a data item that don't change Their Value during The Program Execution.

→ Several Types of C-Const. That are allowed in C-lang. are :-

Good Write

- ① Integer Const. (Whole N.o. Without decimal Part)
- ② Decimal Integer Const. (0-9)
- ③ Octal Integer Const. (0-7) (Octal number system)
- ④ Hexadecimal Integer Const. (0-15) where "10=A, 11=B, 12=C, 13=D, 14=E & 15=F"
- ⑤ Real Const. are N.o. having a fractional part are called Real Const. or floating point const.
- ⑥ Character Const. contains a character and closed with single quotes.

→ Operators in C-language

1. Arithmetic operators

- It is consist of { +, -, *, /, % }.
- % can't be used on float & double operands. In Modulus operator, The sign of result of a%b will be same as sign of 'a'.

$$\text{Ex: } 16 \% 3 = 1 \quad \text{But } 16 \% -3 = 1$$

$$-16 \% 3 = -1 \quad \text{But } -16 \% -3 = -1$$

2. Relational operators

- It is consist of { >, <, >=, <= }.

→ The output of relational operators is always an Boolean Value.

3. Logical operators

- These are the operators which is used to perform logical operations and Expressions.
- It is consist of { &, &&, ||, ! }.

4. Equality operators

- C-language supports 2 kinds of Equality operator To compare The operands for Strict Equality (==) and Inequality (!=).

Good Write

5. Unary operators

→ Unary operators = { $a+$, $a-$, $++$, $--$ }.

6. Ternary / Conditional Operator → operator returns one value when cond "is True" and another when cond "is False"

→ Statement 1 ? Statement 2 : Statement 3 ; → SYNTAX

① eq:- #include < stdio.h >

```
int main() {
```

```
int a, b, c, large;
```

`close();`

11 We can Input Value of a,b & c by prints & Scan ---

printf("large");

3

$$\therefore \textcircled{1} \quad a > b = \text{True} \longrightarrow a > c = \text{True} \longrightarrow \text{large} = a$$

$\rightarrow 0 > c = \text{False} \longrightarrow \text{folge} = c$

$$\textcircled{2} \quad a > b = \text{False} \longrightarrow b > c = \text{True} \longrightarrow \text{folge} = b$$

$b < c = \text{False} \rightarrow \log_b c = c$

7. Bitwise Operators

→ Bitwise OR, Bitwise AND, Bitwise NOT, Bitwise XOR, Shift Operators

→ Bitwise operators performs on Bits.

$$\text{int} a = 27 = (0001\ 1011)_2$$

$$int b = 39 = (0010 \ 0111)_2$$

① Bitwise And ($\&$)

i) Bitwise OR (11)

\rightarrow 0 0011011

$\rightarrow 0001\ 1011$

8 0 0100111 ⚡ period 15.01 0010 0111 wrapped around - 1 ⚡

$$\overline{00000011} = (3.)_{10}$$

$$00\ 11m\ 1111) = (63),_{10}$$

Good Write

③ Bitwise NOT (\sim)

→ Let, Value of Integer a is x , Then, $\text{Not}(x) = -(x+1)$

$$\text{Not}(x) = -(x+1)$$

$$\therefore \text{Not}(27) = -(27+1) = -28 \rightarrow \text{Bitwise NOT of } 27 = -28$$

$$\text{Not}(39) = -(39+1) = -40 \rightarrow \text{Bitwise NOT of } 39 = -40$$

⑤ Bitwise XOR (^)

$$\rightarrow 0001 \ 1011$$

$$\wedge \quad 0010 \ 0111$$

$$\underline{0011 \ 1100} = (60)_{10}$$

⑥ Left Shift ($<<1$)

→ In Left Shift, Remove The Leftmost Bit and Add "0" on Rightmost bit.

$$\textcircled{1} \text{ Eg: } 0001 \ 1011 \xrightarrow[\text{most bit}]{\text{Remove left}} 0011 \ 0110 \xrightarrow{\text{Add } 0 \text{ in last}} 0011 \ 0110$$

$$\therefore a << 1 = (0011 \ 0110)_2 = \boxed{1110} \quad (54)_{10}$$

⑦ Right Shift ($>>1$)

→ In Right Shift, Remove The Rightmost Bit and Add "0" on Leftmost Bit.

$$\textcircled{2} \text{ Eg: } 0010 \ 0111 \xrightarrow[\text{most bit}]{\text{Remove right}} 001 \ 0011 \xrightarrow{\text{Add } 0 \text{ on beginning}} 0001 \ 0011$$

$$\textcircled{3} \text{ Eg: } 0010 \ 0111 \xrightarrow[\text{most bit}]{\text{Remove right}} 001 \ 0011 \xrightarrow{\text{Add } 0 \text{ on beginning}} 0001 \ 0011 = (19)_{10}$$

$$\therefore \text{Thus, } a \& b \rightarrow 3 \quad a \oplus b \rightarrow 63 \quad a + b \rightarrow 64 \quad a - b \rightarrow 0$$

$$a \& b \rightarrow 63 \quad a + b \rightarrow 64 \quad a - b \rightarrow 0$$

$$\sim a \rightarrow -28 \quad \text{JAI LOGIC!!!}$$

$$(a \& b) \sim b \rightarrow -40$$

$$(a \& b) \sim a \rightarrow 60$$

$$a << 1 \rightarrow \boxed{54}$$

$$b >> 1 \rightarrow 19$$

⇒ Comma Operator → Special Operator

- % in "%d" is used to print an Variable and d represent Int datatype.
- It is an special operator in "C" which evaluates the 2nd operant and discards the result of the 1st one. The second operant is evaluated and result of same is returned.

int a = (19, 30) = 30;

⇒ Assignment Operator (=)

- This operator assign value to Variable.

① Assignment Operator = { + = , - = , * = , / = , % = }.

* sizeof() operator is used to check the size of an Variable / Entity. sizeof() operator is an unary operator which is used to calculate the size of datatype.

② Eg:- "%3d" → Will Input Integer of 3 Byte

"%2s" → Will Input Variable of size - t.length

Operator Precedence :::: (*, /, %) > (+, -)

"S.M.P." We will firstly solve the set (*, /, %) from left

To Right and Then will solve (+, -) from left To right.

① Eg:- int a = 3, b = 5;

Solve::: a + = b * 4 - 7

$$a = a + b * 4 - 7$$

$$a = 3 + 5 * 4 - 7 \quad (\text{As, operator precedence of } * \text{ is more}).$$

$$a = 3 + 20 - 7 = 10 \rightarrow \text{done}$$

② Eg:- int a = 3 * 4 / 5 % 6 + 6 - 4 (left To Right for *, /, %)

$$\text{int a} = 12 / 5 \% 6 + 6 - 4 \quad (\text{left To Right for } *, /, \%)$$

$$\text{int a} = 2 \% 6 + 6 - 4 \quad (\text{left To Right for } *, /, \%)$$

$$\text{int a} = 2 + 6 - 4 = 4$$

Good Write

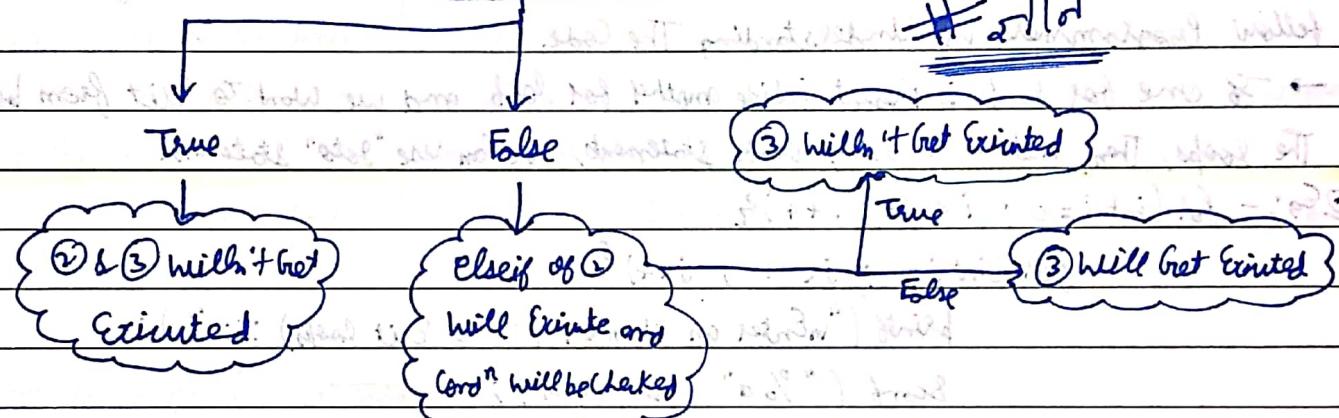
⇒ If Statement

→ SYNTAX :::: if (Condⁿ) { mnm } ① ; Then If condition is True
 else if (Condⁿ) { mnm } ② ; Then If condition is False
 else { mnm } ③ ;

Note → Because of elseif, If The Condⁿ ① is True, Then, Condⁿ ② & ③ will not Execute.

If We used Else instead of elseif in ②, Then, If Statement ① is False, Then, ② & ③ will Get Executed, Thus, To Prevent This, We use elseif instead of else.

~~if (Condⁿ of ①) and if (Condⁿ "else") will get executed~~



⇒ Switch Statements

→ SYNTAX :::: switch (Variable){
 Case Value1: mnm; In Each Case, don't forget to add break
 Case Value2: mnm; folget to add break
 ...
 Case ValueN: mnm; One Case is True, Rest below
 default: mnm; Cases will also get Executed }

Q :- switch (age){
 Case 16: print ("You are in Class 10");
 ...
 Case 18: print ("You are in Class 12");
 default: print ("You are Not in Board class");
 break;

Good Write
3

break;