1. **What is the meaning of assembly language?**

**Assembly language and assemblers,**

An assembly language is a low-level [programming language](https://techterms.com/definition/programming_language) designed for a specific type of [processor](https://techterms.com/definition/processor). It may be produced by [compiling](https://techterms.com/definition/compile) source code from a high-level programming language (such as [C/C++](https://techterms.com/definition/cplusplus)) but can also be written from scratch. Assembly code can be converted to machine code using an [assembler](https://techterms.com/definition/assembler).

Since most [compilers](https://techterms.com/definition/compiler) convert [source code](https://techterms.com/definition/sourcecode) directly to machine code, software developers often create [programs](https://techterms.com/definition/program) without using assembly language. However, in some cases, assembly code can be used to fine-tune a program. For example, a programmer may write a specific [process](https://techterms.com/definition/process) in assembly language to make sure it functions as efficiently as possible.

While assembly languages differ between processor [architectures](https://techterms.com/definition/architecture), they often include similar instructions and operators. Below are some examples of instructions supported by [x86](https://techterms.com/definition/x86) processors.

MOV - move data from one location to another

ADD - add two values

SUB - subtract a value from another value

PUSH - push data onto a [stack](https://techterms.com/definition/stack)

POP - pop data from a stack

JMP - jump to another location

INT - interrupt a process

The following assembly language can be used to add the numbers 3 and 4:

mov eax, 3  - loads 3 into the register "eax"  
mov ebx, 4  - loads 4 into the register "ebx"  
add eax, ebx, ecx  - adds "eax" and "ebx" and stores the result (7) in "ecx"

In c programming:

Int sum = 3 + 4;

Writing assembly language is a tedious process since each operation must be performed at a very basic level. While it may not be necessary to use assembly code to create a computer program, learning assembly language is often part of a [Computer Science](https://techterms.com/definition/computer_science) curriculum since it provides useful insight into the way processors work.

1. **Why c program is called procedural language?**

Because C programs follow a procedure of steps written in it, called functions. It follows a top-down approach i.e. much importance is given to flow of program rather than on data on which functions operate.

Functions are also referred to as procedures.

On the other hand, Java/C++ are object oriented languages. They have a bottom up approach. They revolve around classes and objects, and follow principles of inheritance, abstraction, encapsulation & polymorphism.

C language doesn't have any such functionality. It follows whatever steps are written in a function. Hence, the name.

1. **What is constant? Not understanding the program  arithmetic operator,  %, ++, --?**

#include<stdio.h>

int main() {

  const int email\_length = 80;

  email\_length++;

  printf("Sum of email\_length = %d", email\_length);

}

#include<stdio.h>

int main() {

// uncomment the commented lines and vice versa

//Database constraint us not to take input beyond 80 char long email

int email\_length = 80;

// this is also equal to

email\_length = email\_length + 1;

//email\_length++; //increment email\_length by 1

email\_length = email\_length -1;

//email\_length--;

printf("%d\n", email\_length);

/\*Comment: Example of % operator\*/

// Comment: Find the remainder when 80 is divided by 5

int remaindr = 80 % 3;

printf("%d", remaindr);

}

1. **Bitwise operators  program confusing.**

Skip, not used in daily life, not very relatable. Integers are expressed in decimal values, how will you visualize the bits like shown below?

**12 = 00001100 (In Binary)**

**25 = 00011001 (In Binary)**

**Bitwise AND Operation of 12 and 25**

**00001100 – (12)**

**|00011001 – (25)**

**\_\_\_\_\_\_\_\_**

**00001000  = 8 (In decimal)**

Now how will you perform AND over the bits not decimal, for that we have bitwise operators. Bit operations over decimal without converting decimal numbers into their corresponding bits.

#include<stdio.h>

int main() {

// AND between binary forms of 12 and 25

// without explicit conversion into their binary forms

int a = 12, b = 25;

printf("Output = %d", a&b);

return 0;

}

1. **Ternary operator and conditional operator are same?**

Yes

A picture containing text

Description automatically generated

int a = 10, b = 20, c;

if (a < b) {

c = a;

}

else {

c = b;

}

printf("%d", c);

Syntax

**condition ? value\_if\_true : value\_if\_false**

int a = 10, b = 20, c;

c = (a < b) ? a : b;

printf("%d", c);

Program:

#include<stdio.h>

int main() {

// find min of a and b

int a = 10, b = 20, min;

if (a < b) {

min = a;

}

else {

min = b;

}

printf("Usual way %d\n", min);

// condition ? value\_if\_true : value\_if\_false

int conditional\_min;

//shortform to write simple if/else clause

conditional\_min = (a < b) ? a : b;

printf("conditional statement min %d\n", conditional\_min);

//find the max value out of a and b

int conditional\_max;

conditional\_max = (a > b) ? a : b;

printf("Ternary operator to find max between 10, 20 and max is %d", conditional\_max);

}

1. **Storage bits in floating point? Long double?**

Value of pi (11 significant digits)

3.14159265359 =

314159265359((absolute value) \* E-11 (exponent)

In the memory your value will be stored something like this

= [absolute value][ exponent] = [314159265359] [-11]

1. **How to work with c  program  in  job place?**

Program which executes fast

Helps to understand the fundamentals of Computer Theories, such as Computer Networks, Compiler Designing, Computer Architecture, Operating Systems are based on C programming language.

IMPORTANT:

1. You should be able to write a simple code.
2. Have good understanding of pointers (Memory management and usage – lower level concepts – little tough to understand but worth the effort).
3. **Can we use keywords in variable declaration?**

No

1. **Difference between implicit and explicit declaration.**

Explicit means declaring variable like in c. Implicit declaration in variable declaration in python.

1. **Brief explanation of  "Mics operator"**

Next classes

1. **Distinguish between malloc() & calloc() memory allocation. .**

Pointers class. (Malloc function is used to allocate a single block of memory space while the calloc function is used to allocate multiple blocks of memory space. Each block allocated by the calloc function is of the same size.)

1. **What is keyword auto for? ...**

**auto**: This is the default storage class for all the variables declared inside a function or a block. Hence, the keyword auto is rarely used while writing programs in C language. Auto variables can be only accessed within the block/function they have been declared and not outside them (which defines their scope).

**13. What are the valid places for the keyword break to appear. ...**

Breaking the loop. Switch, next class

**14. Explain the syntax for for loop. ...**

Next class

**20. What is difference between including the header file with-in angular braces < > and double quotes “ “**

Inbuilt and self-created one.

Bookmarks:

<https://www.geeksforgeeks.org/c-programming-language/>

<https://www.tutorialspoint.com/cprogramming/index.htm>