First computer programmer : **Augusta Ada King-Noel, Countess of Lovelace**

**1815 - 1852**

was an English mathematician and writer, chiefly known for **her work on Charles Babbage's proposed mechanical general-purpose computer, the Analytical Engine.** She was the first to recognise that the machine had applications beyond pure calculation, and **published the first algorithm** intended to be carried out by such a machine. As a result, she is sometimes regarded as the first to recognise the full potential of a "computing machine" and the **first computer programmer.**

**History of programming languages :**

Assembly Programming languages

.

.

1963 - CPL (Combined Programming Language) - Designed by Christopher Strachey.

(CPL was also nicknamed by some as "Cambridge Plus London"[2] or "Christopher's Programming Language").

1967 - BCPL - basic combined programming language/ before c programming language, Designed by Martin Richards

1969 - B programming language - Designed by - Ken Thompson Developer, Ken Thompson, Dennis Ritchie

B was derived from BCPL, and its name may be a contraction of BCPL.

Thompson's coworker Dennis Ritchie speculated that the name might be based on Bon,

an earlier, but unrelated, programming language that Thompson designed for use on Multics(os).

C : First appeared 1972

Designed by Dennis Ritchie

Developer Dennis Ritchie & Bell Labs

**Influenced by :**

B (BCPL, CPL), ALGOL 68, Assembly, PL/I(programming language one), FORTRAN

**Influenced**

Numerous: AMPL, AWK, csh, C++, C--, C#, Objective-C, BitC, D,

Go, Java, JavaScript, Julia, Limbo, LPC, Perl, PHP, Pike, Processing, Python,

Ring, Rust, Seed7, Vala, Verilog (HDL)

**Executing C program from command prompt/terminal**

1. Create file Hello.c
2. Open terminal/command prompt
3. Compile command : **gcc Hello.c**
4. Executable file **a.out** will be generated
5. Command **./a.out** will display output

It has been closely associated with the UNIX system where it was developed, since both the system and most of the programs that run on it are written in C.

C is called a "system programming language" because it is useful for writing compilers and operating systems, it has been used equally well to write major programs in many different domains.

C deals with the same sort of objects that most computers do, namely characters, numbers, and addresses.

Inner functions not allowed.

The functions of a C program may exist in separate source files that are compiled separately.

Variables may be internal to a function, external but known only within a single source file, or visible to the entire program.

A preprocessing step performs macro substitution on program text, inclusion of other source files, and conditional compilation.

C is a relatively "low level" language.

High level languages are designed for the ease of the person writing the language.

Lower level languages are designed for the ease of the computer running the language.

C provides no operations to deal directly with composite objects such as character strings, sets, lists, or arrays(crud).

No automatic memory management like garbage collector.

Most C implementations have included a reasonably standard

collection of such functions.(stdio.h, string.h, stdlib.h , errono.h)

C offers only straightforward, single-thread control flow :

Tests, loops, grouping, and subprograms, but not multiprogramming, parallel opera-

tions, synchronization.

For many years, the definition of C was the reference manual in the first

edition of The C Programming Language(issue : conflicting syntax). In 1983, the American National Standards Institute (ANSI) established a committee to provide a modern,

comprehensive definition of C. completed in 1988.

C is not a strongly-typed language, but as it has evolved, its type-checking

has been strengthened.

Compilers will warn of most type errors, and there is no automatic conversion of incompatible data types.

Ex : char to int possible, char to bool not possible