**Chapter 1 - Introduction**

**Main languages:**

i) Machine language:

Consists of 0s and 1s. It is too slow, tedious, and error-prone.

ii) Assembly language:

Consists of english-like abbreviations. **Assemblers** are used to convert the assembly language into machine language.

iii) High-level language:

In this, a single statement executes considerable amount of tasks. **Compilers** are used to convert high-level language to machine language. Compilation takes time but execution takes less time. **Interpreters** are used to execute the high-level languages directly without converting into machine language (i.e. compilation). So, Interpreters take less time in compilation and more time in execution when compared to Compilers.

**Six phases of execution:**

a) Edit:

Writing a C program in a text editor which is saved and can be edited later with corrections. It is stored in the Secondary storage (i.e. disc) with ".c" extension.

b) Preprocess:

In a compiler, 'preprocessing' before the compilation. Preprocesser commands called '*preprocesser directives'* include other files for compilation.

c) Compile:

The C code is converted into Machine language code.

d) Link:

The machine language code consists some holes - references to functions defined in the C standard library or some private library. The **Linker** linksmachine language code with the missing function.

e) Load:

f) Execute