

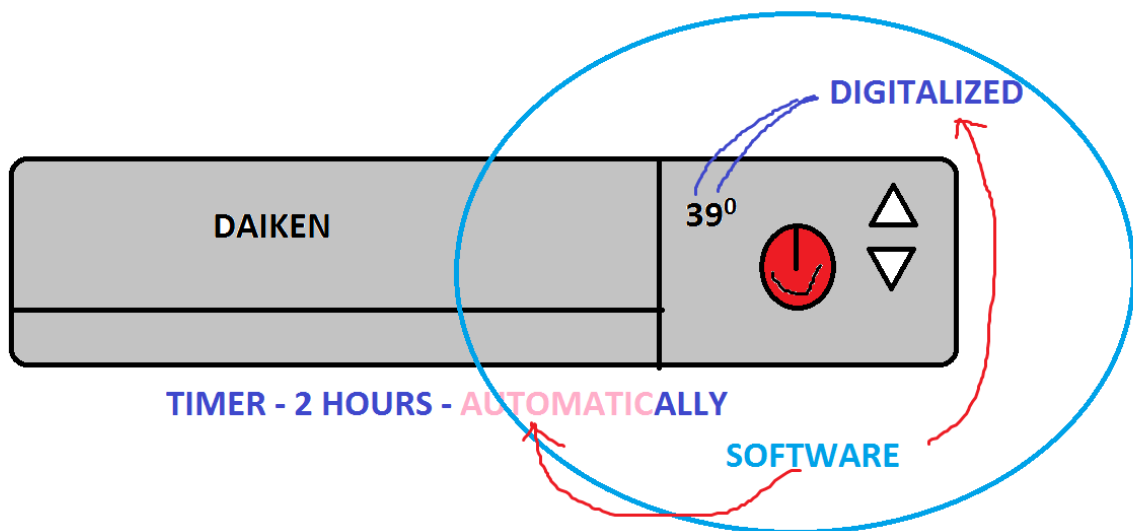
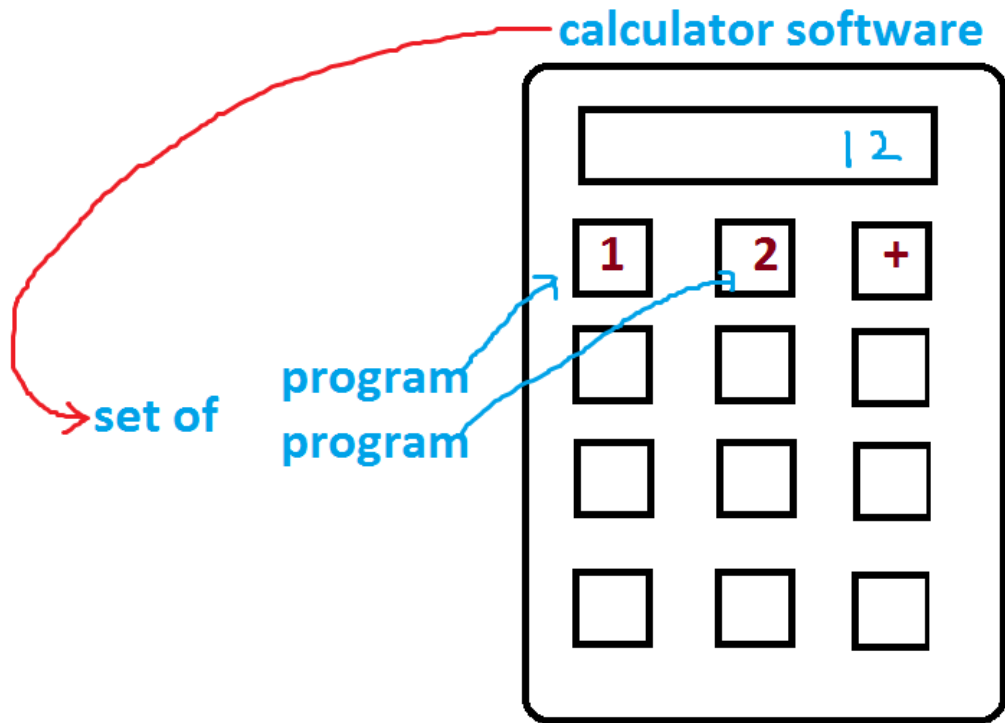
INTRODUCTION TO C LANGUAGE

C is a

1. It is a high level / middle level programming language.
2. It is a compiler based programming language.
3. It is a procedure oriented programming language [POP]
4. It is a general purpose programming language.

What is a program? Set of instructions are called program.

What is a software? Set of programs are called software. As per IT Industry software is a digitalized and automated process.



Basically software divided into 2 types.

1. System software

The system is used to manage the system hardware and provides the interface between the hardware and user.

Eg: operating system, device drivers, translators

2. Application software: It is designed for a particular task.

Eg: whatsapp, paytm, irctc, vlc media player,

What is a language?

Generally the languages like telugu, English, hindi, Marathi are called human languages, which are used to communicate with the humans.

C / C++ / Java / .Net / Py are called computer programming languages and they are used to write the programs [software] to communicate with the machines.

C Language
↓
set of Instructions
↑
set of programs
↑
software
↑
software engineers

wipro/tcs/infosys
mnc companies



```
Line 6    Col 1    Insert Indent Tab Fi
#include<stdio.h>
void main()
{
    int i;
    for(i=1;i<=100;i++)
    printf("%4d",i);
}
```

Basically the computer languages divided into 3 types.

1. **Machine language / binary code [0,1]**: Created with binary code.
Eg: 10001111
2. **Low level language / assembly language**: Created with English like shortcuts called **MNEMONICS**.
Eg: gd mrg, sub, add,
3. **High level language**: Created with simple English and easy to understand.

Eg: good morning, subject, addition,

Asap – as soon as possible

C is a high level with low level features. Hence it is a **middle level language**.

C **low level features** used to develop **system software** and **high level features** used to develop **application software**. Hence C is called it is a multi-purpose programming language.

Example for low level / assembly language:

The screenshot displays the Turbo C++ IDE interface. The top menu bar includes File, Edit, Search, Run, Compile, Debug, Project, Options, Window, and Help. The title bar reads "Turbo C++ IDE". The main editor window, titled "ASS.CPP", contains the following C code:

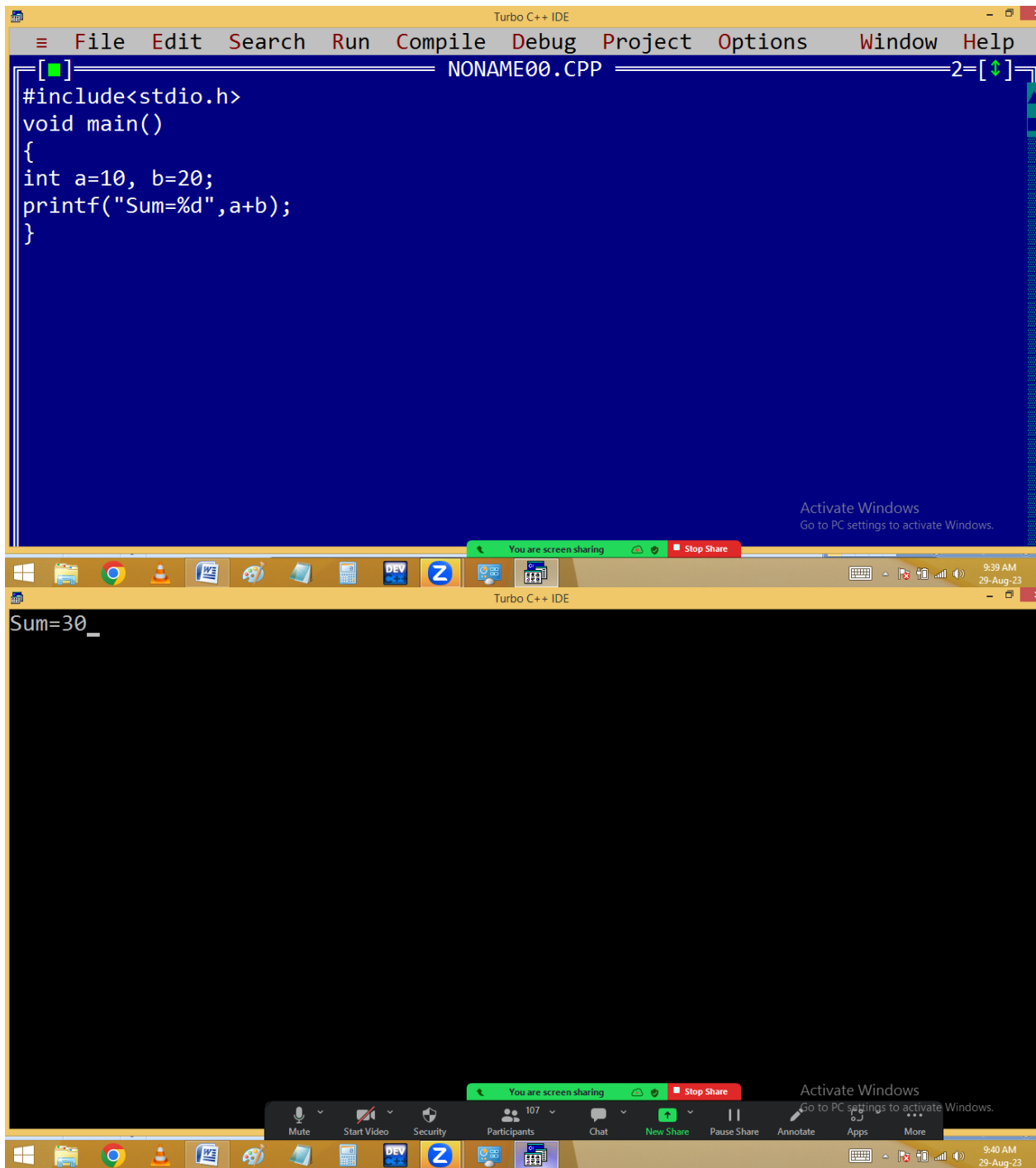
```
#include<stdio.h>
void main() {
    int a = 10, b = 20, c;

    asm {
        mov ax,a
        mov bx,b
        add ax,bx
        mov c,ax
    }

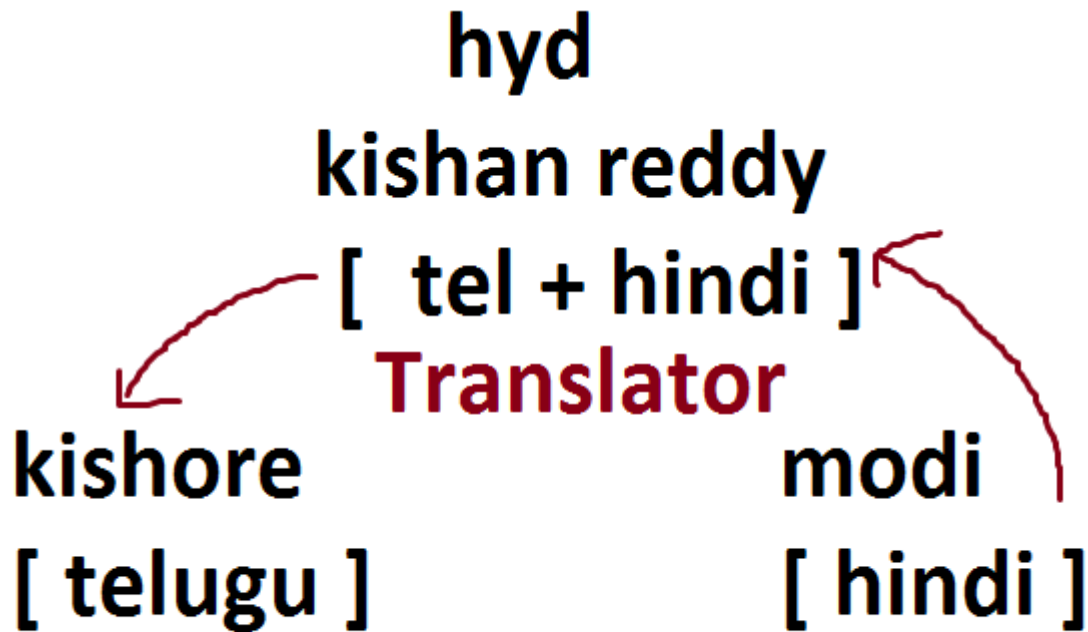
    printf("c= %d",c);
}
```

The status bar at the bottom of the editor shows "9:13". Below the editor, a black console window displays the output "c= 30". The Windows taskbar at the bottom shows various application icons, a system clock indicating "9:33 AM 29-Aug-23", and a notification area with "You are screen sharing" and "Stop Share" buttons. A watermark "Turbo C++ IDE" is visible in the center of the console window.

Example for high level program:



What is a translator?

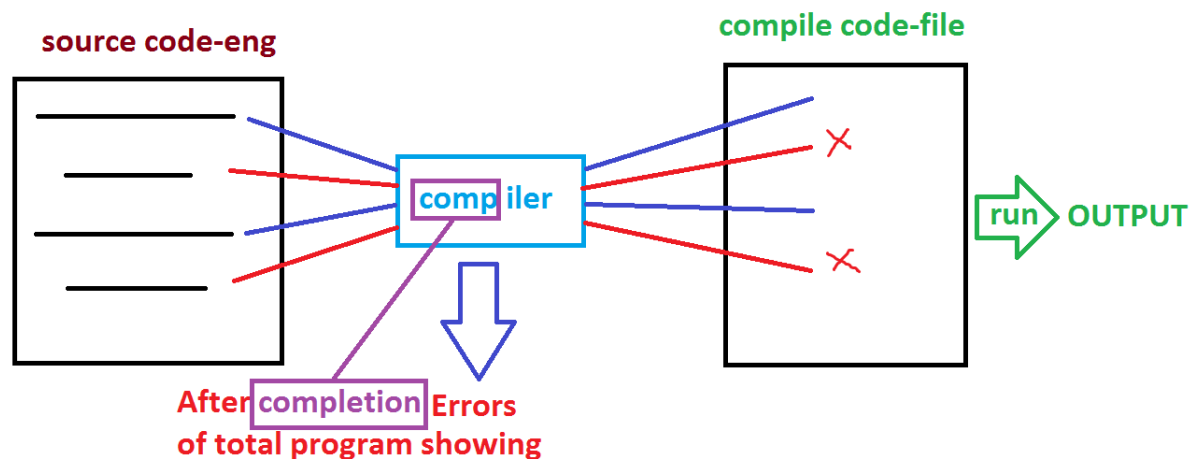


Always the user given instructions are English, which is called source code / source program. But the computer understandable code is binary code / machine language. **to convert this source code into binary code and to check the errors** we are using the translators like

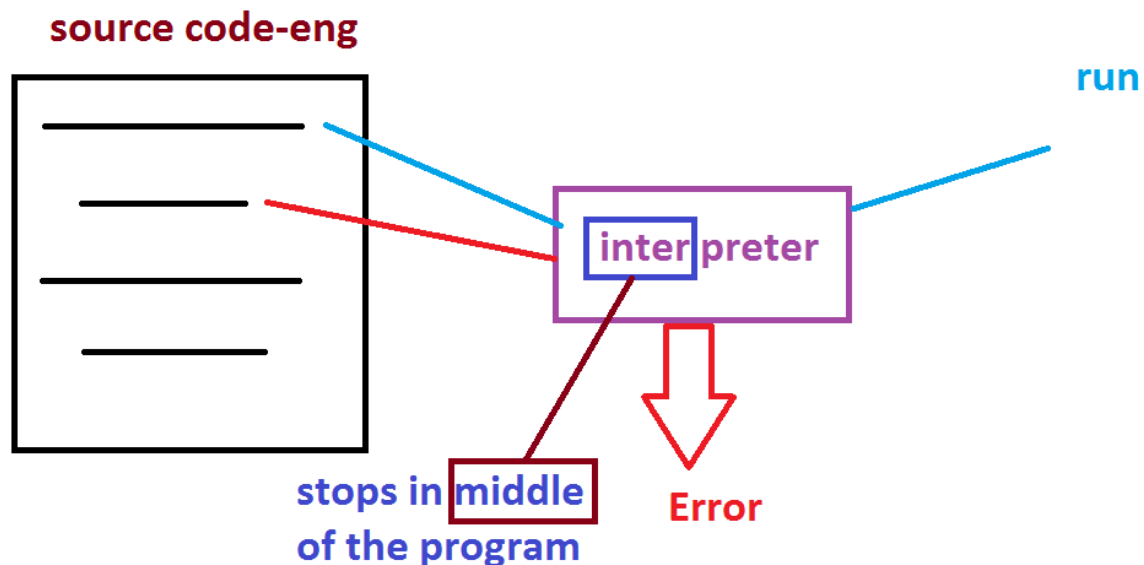
1. Compiler
2. Interpreter
3. Assembler

Compiler and interpreter both used to convert high level programs to machine language.

Compiler converts the whole source code into binary code **at once** by leaving error lines.



Interpreter converts **line by line**.



Assembler is used to convert our low level programs / assembly language to machine language.

Assembler working style is similar to the compiler.

In C we are using compiler as a translator. Hence it is called compiler based programming language.

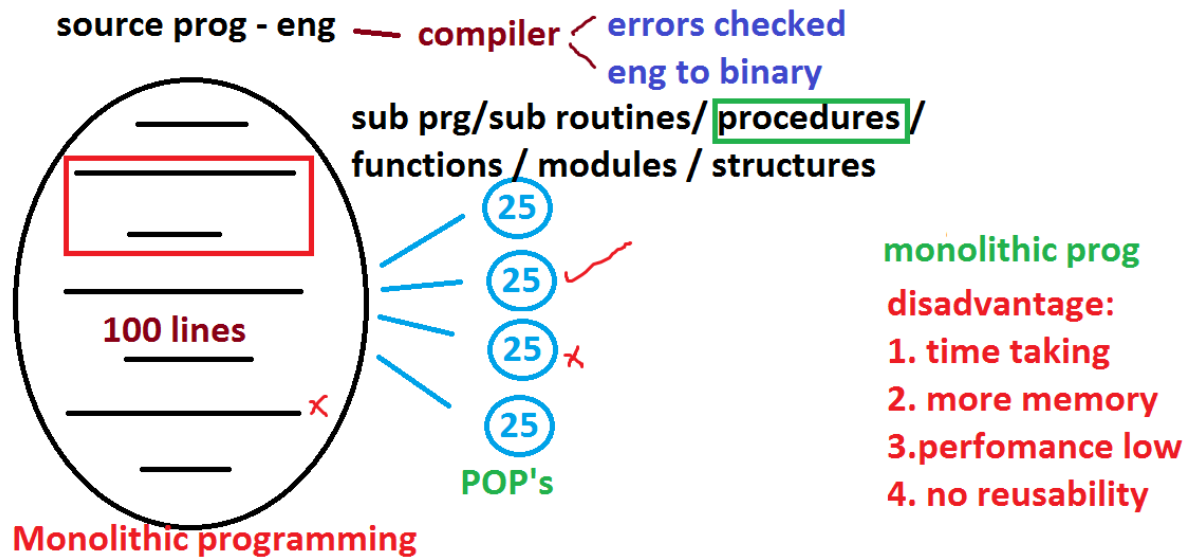
In java / .net / python we are using compiler and interpreters. Hence they are called compiler based interpreted programming languages.

What is programming paradigm?

Every programming language follows a particular structures with rules and regulations and this is technically called programming paradigm.

Before C language, the languages are using monolithic programming paradigm.

In monolithic programming paradigm, the entire application designed with single program.



Example for monolithic programming:

The image shows a screenshot of a Turbo C++ (TC) IDE. The top window displays the source code of a C program. The code is as follows:

```
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 76 Insert Indent Tab Fill Unindent * E:NONAME.C
#include<stdio.h>
void main()
{
printf("-----\n");
printf("Good morning to all\n");
printf("-----\n");
printf("Welcome to C\n");
printf("-----\n");
printf("THANK YOU\n");
printf("-----");_
}
```

The bottom window shows the output of the program, which matches the printf statements in the code:

```
-----
Good morning to all
-----
Welcome to C
-----
THANK YOU
-----
```

Both windows have a taskbar at the bottom with various application icons and a system clock showing 10:29 AM on 30-Aug-23. A green status bar at the top of the bottom window indicates "You are screen sharing".

To avoid the disadvantages in monolithic programming, they have introduced POP's in C language. in C a big program divided into several small sub programs /

subroutines / procedures / functions / structures / modules. Hence C program is collection of procedures, it is called procedure oriented programming structure.

POP Advantages:

1. **Modularity**: Dividing a big program into small modules based on program requirement.
2. **Simplicity**: Easy to read and understand.
3. **Reusability**: write once, use many times.
4. **Efficiency**: performance is high

Example for procedure oriented programming [pop]:

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays the source code of a C program. The code defines a function `line()` that prints a dashed line, and a `main()` function that calls `line()` and prints three messages: "Good morning to all", "Welcome to C", and "THANK YOU". The bottom window shows the output of the program, which displays the three messages, each preceded by a dashed line. The IDE interface includes a menu bar with options like File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. A status bar at the bottom indicates "You are screen sharing" and "Stop Share".

```
File Edit Run Compile Project Options Debug Break/watch
Line 15 Col 8 Insert Indent Tab Fill Unindent * E:NONAME.C
#include<stdio.h>
void line()
{
printf("-----\n");
}

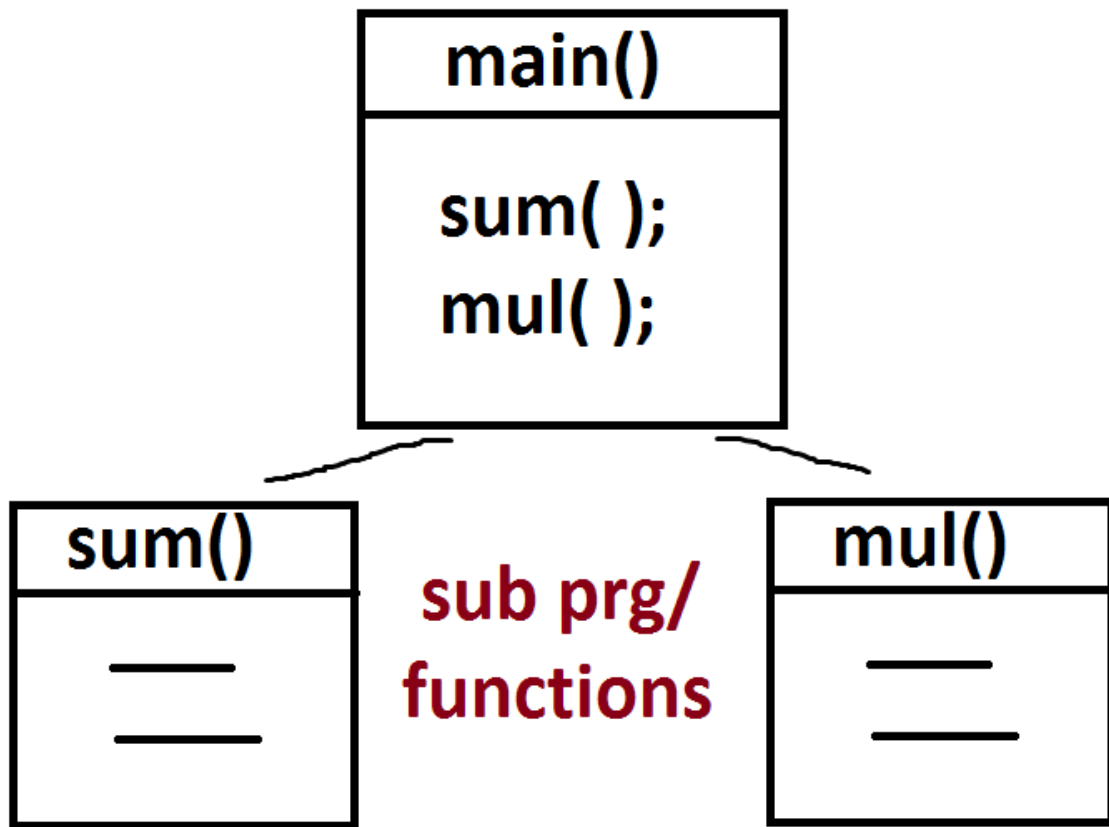
void main()
{
line();
printf("Good morning to all\n");
line();
printf("Welcome to C\n");
line();
printf("THANK YOU\n");
line();
}
```

Good morning to all

Welcome to C

THANK YOU

Function oriented programming structure:



C is a procedure oriented programming language.

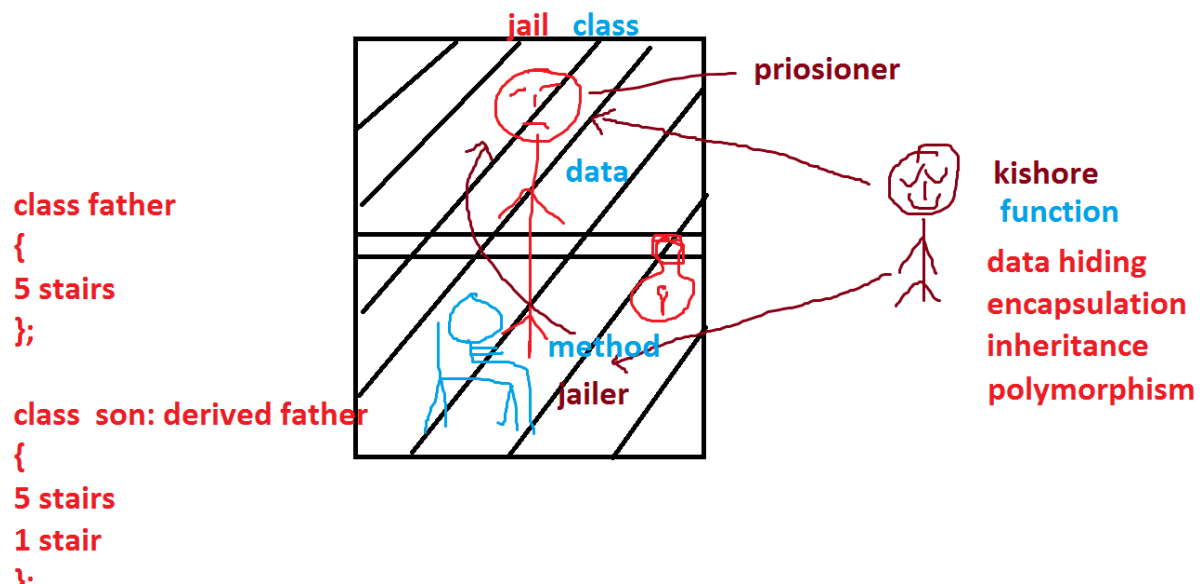
C++ & Python – **Multi paradigm programming languages** because of they support both pop & oop.

Java / .net – OOP's

Disadvantages of pop:

Not secured.

In OOP's:



Why C is a general purpose language?

Using C language we can develop the applications like

1. Operating system

Eg: windows, mac, android, linux,...

2. Editors

Eg: Notepad, wordpad, Ms-Word,....

3. Translators

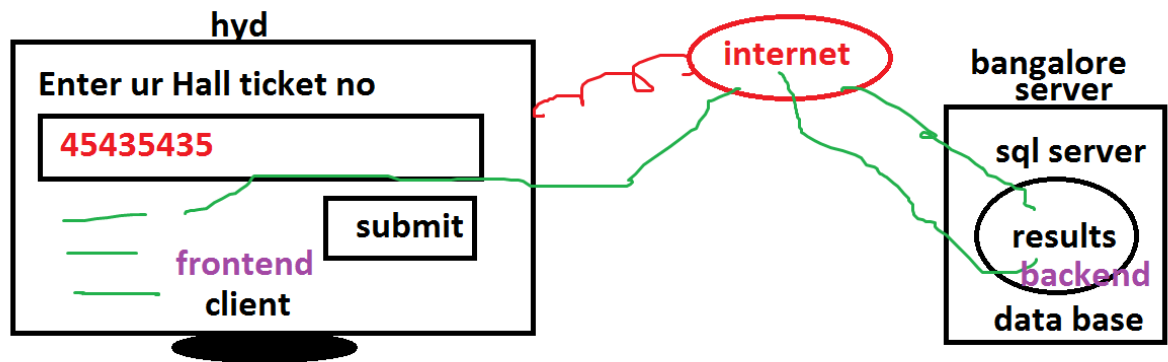
Eg: compiler, interpreter, assembler

4. Commercial applications

Eg: super market / hotel / college programs

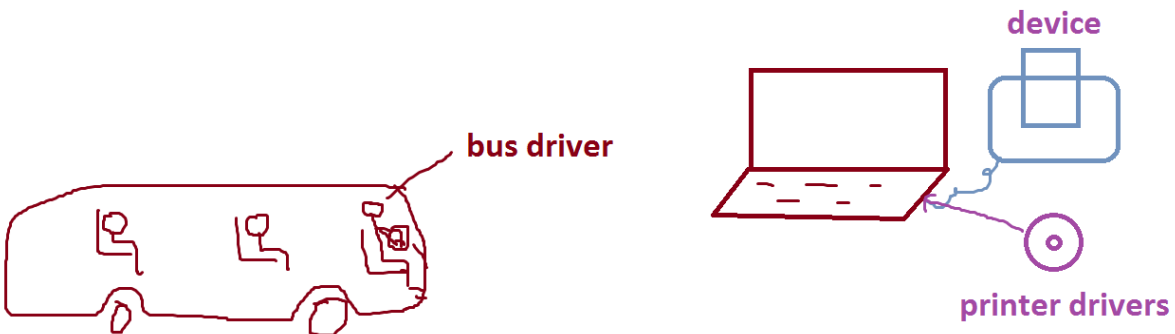
5. Data base

Eg: Oracle, SQL Server, My SQL, Mongoddb, SQLite,...

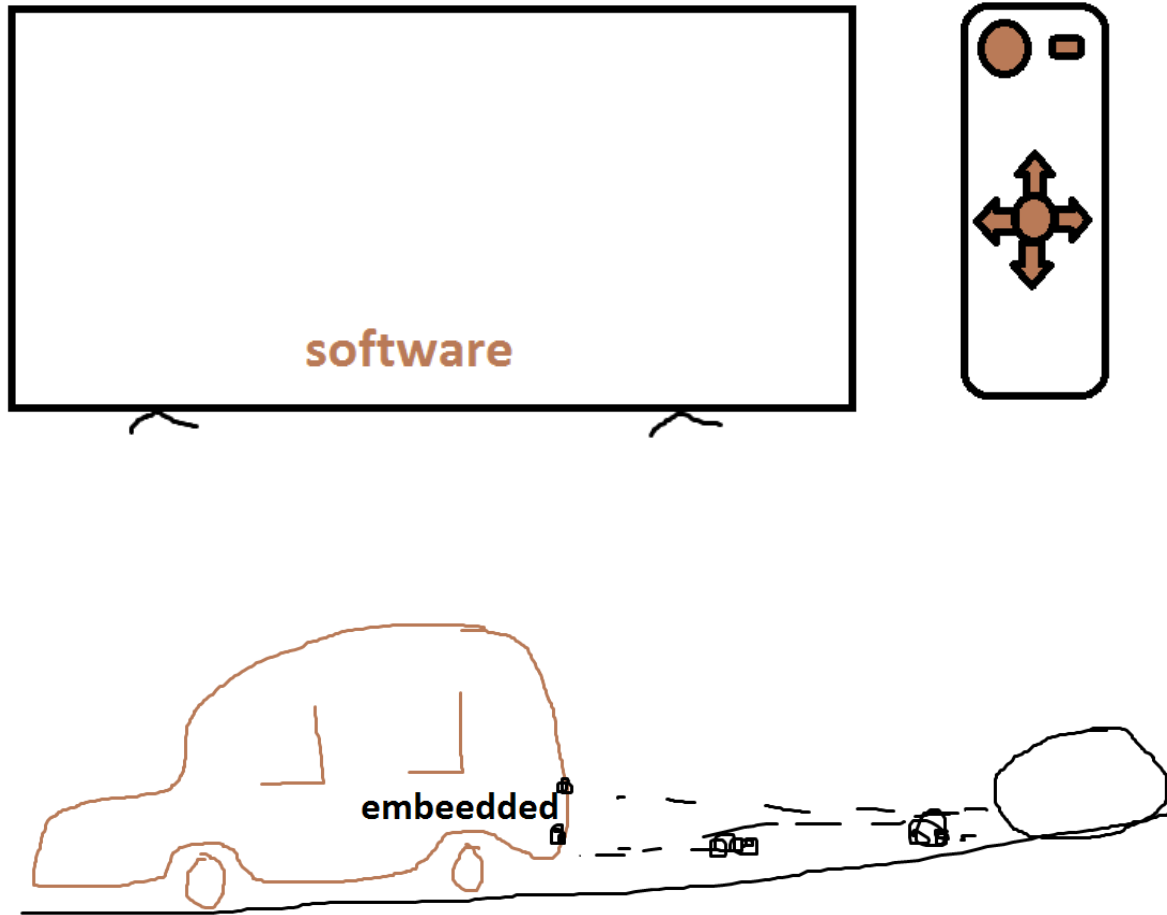


6. Device drivers

Eg: audio / video / printer / usb drivers,....



7. Embedded applications



8. antivirus

eg: Norton, avast, macfee,

9. media players

eg: VLC, mx player, windows media player,...

10. browsers

eg: chrome, firefox, edge,...

11. PC & Mobile games

12. Stand alone applications

Due to all these features c is also called it is a multipurpose programming language.

HISTORY OF C

Basically C language introduced in **1972** by “**DENNIS RITCHIE**”, software engineer in **AT & T** [American Telephone and Telegraph] **Bell labs**, located at Murray Hills, New Jersey, USA.

Richie adopted C language from **B language**, designed by “**KEN THOMSON**”, Software engineer in AT & T Bell labs.

Thomson adopted B language from BCPL [Basic Combined Programming Language, designed by “**MARTIN RICHARDS**”, assistant professor in Cambridge University.

In **1989 ANSI** [American National Standards Institute] released a new version of c language with the name “**C-89**” / “**ANSI-C**”.

In **1999 ISO** [International Standard Organization] released a new version of c language with the name

“C-99”.

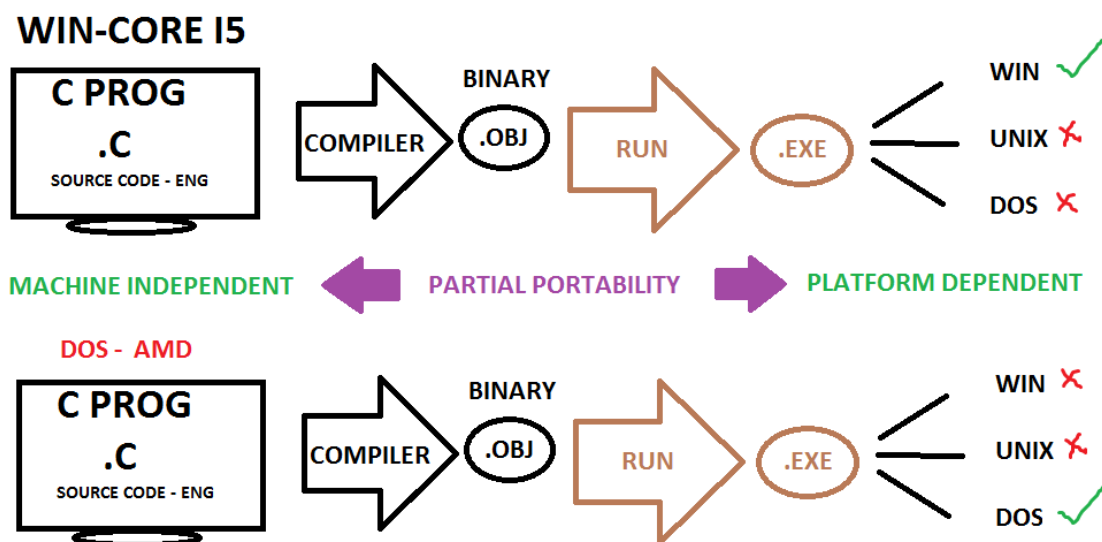
Basically C language developed to rewrite **UNIX** operating system.

Nowadays we can create and execute a c program on any machine with any processor. Hence c is called it is a **machine independent programming language**. i.e. we can create and execute c program on 80386 / 80486 / Pentium / intel core i3 / i5 / i7 / i9 / i11 / amd rayzon etc processors.

The languages like **8086** / **8088** are working only on **8086** / **8088** processors. Hence they are called machine dependent programming languages.

C is a platform dependent programming language. i.e. The c program created for one operating system is not working in other operating systems. For example c program created for windows is not working in unix. Due to this problem using C we can't develop web applications. Hence we can develop only the standalone applications with C language. The application installed in a single system and working from only that system is called standalone application.

The languages like Java / .Net / Python are used to develop web applications because of these are **platform independent programming languages**. The web applications installed in a server and accessed from client systems. Hence java / .net / python are called portable languages and c & C++ are called partial portable languages.



MNP - MOBILE NO PORTABILITY



JAVA - WORA - WRITE ONCE RUN ANYWHERE

