GCC:

* GNU project c and c++ compiler
* When you invoke GCC, it normally does preprocessing, compilation, assembly and linking
* If we use -g the size of the application increase.
* -> GCC commands (learn- -c,-s,-e,-g,-pg,-warn,-Idir,-ldir,-Dmacro,-foption,….)
* -ansi: to meet standards
* -std-mentions which standard or version I'll be using.
* -I : to include a particular directory(third party files)
* -c : compiles, -E : preprocessed state
* Gcc -Wall -c main.c - to compile
* Gcc -S main.c-to get assembler code
* Gcc --- save-temps : to get intermediatory files

Make:

* GNU Make utility to maintain groups of programs

-> to avoid re declaration we can keep #ifndef name

-> declaring function prototypes or prototyping of functions:

Int add (int,int)

->In one statement to reach the source file : cd/src

->**Double quoted header files** search for header files which is defined in it.( current directory)

->**Angular header** files checks in currrent directory also in gcc.

-> gcc -c ./src/calc.c

gcc -c ./src/calc.c -I./inc - to go parent directory

-> fatal error:calc.h: No such file or directory

gcc -c ./src/calc.c -I./inc -o ./obj/calc.o-

Make: move the root directory to defined directory

* Created directory structure
* Cd mainprj
* Mainprj>( create sub directories) mkdir inc obj scripts lib src bin
* Created header file(with vi editor)
* Before that
* Cd inc
* Mainprj/inc>vi calc.h
* Cd ../ or cd src (we can move to source file)
* mainPrj/src/> vi calc.c
* mainPrj/src/> vi main.c
* Coming back to main directory - cd../
* mainPrj/>gcc -c ./src/calc.c -I./inc/ -o ./obj/calc.o (convert to object file)
* gcc -c ./src/main.c -I./inc/ -o ./obj/main.o
* Tree
* Vi Git ignore is for ignoring files or folders which is dfined in git ignore file to be upload on cloud platform.
* Git status tell you have added or updated to your local repository

<https://github.com/bhimatak/CGBatch20Oct2024.git>