Agenda

- Revise
- Constant
- Mutable
- Static
    - data member
    - member function
- Reference
- pass by value,address and reference

Constant (demo01)

- once initialized we cannot change its value.

Constant Data Member (demo02)

- If we make the data member as constant then, its value once initialized cannot be changed.
- however we cannnot initialize the data members of the class inside the ctor body.
- they must be initialized inside ctor members initializer list.

Constant Member Functions(demo03)

- If we make the member functions as constant then we cannot moodify the values of non constant data members inside these functions.
- We can make all the display functions and the Inspector functions as constant.

Constant Object (demo04)

- If we make the object as constant then we cannot change the state of an object once it is initialized.
- constant objects can call only constant member functions

Mutable (demo05)

- It is a type qualifier which is used for the non constant data members.
- It is used to modify/mutate the values of non constant data members inside constant member functions.

Static Data Members (demo06 and demo07)

- It is a data member that is designed to shared across muliple objects

- It gets memory allocated on data section.
- all the objects will access the same static data members from this data section.
- they must be initialized outside the class using name of class and scope resolution operator.

static member Function (demo08 and demo09)

- if we want to call the member functions if the class without cretaing the object then such member fucntions should be made as static.
- static member functions are designed to be called on classname using scope resolution operator.
- static member functions do not get this pointer.
- static member functions can access only static data members, they cannot access non static data members of the class.

Reference (demo10 & demo11)

- it is an alias for an existing memory location
- It does not allocate any seperate memory like the pointers.
- no confusion of address and value (dereferencing) like ptr.
- It is recommended in cpp to use pass by reference whereever possible.

Friend Fuction (demo12)

- It is a non member function which is declared as a friend inside the class.
- It is designed to access the private members of the class on class object inside them.