Agenda

- Reference
- Shallow & Deep Copy
- Copy Ctor
- Static
- Exception handling

Reference (demo01.cpp)

- It is just an alias given to an existing memory location
- For reference their is no seperate memroy allocation like pointer
- It is just anaother name of that address.

Shallow copy,Deep Copy & Copy Ctor (demo02.cpp to demo03.cpp)

- If you try to copy the previously cretaed object into newly created object then your copy ctor gets called
- In every class default copy ctor exixts which does the shallow copy
- shallow copy work fine with the normal data members.
- If you class consists of pointer type of data members and dynamic memroy allocation is done then shallow copy is not going to work.
- Solution to it is to perform deep copy.
- To perform deep copy you have to write your own copy constructor

Static Data Members(demo05.cpp)

- the data member sthat are designed to share between muliple objects
- the static data memebrs need to be initialized outside the class using class name and scope resolution operator

Static Member Functions (demo06.cpp)

- If you want to call the member functions of the class without cretaing the object of the class then make such functions as static
- static member functions do not get this pointer
- inside static member functions you can access only static data memebers, you cannot access nono static data members
- to call the static fucntions you hae to use the class name and scope resolution operator

Exception handling (demo07.cpp & demo08.cpp)

- try
- catch
- throw
- It is done to seperate Business Logic and Error Handling Logic
- when exeception is thrown then their must exist a catch block that mataches the type of exception that is thrown
- for every try block you can have multiple catch blocks
- if you want to handle all the execptions in single catch block then handle it inside generic catch block
- generic catch block should be the last catch block of the catch block series