Agenda

- Static
  - data member
  - member function
- Reference
- pass by value,address and reference
- Dynamic memory allocation in C and C++.
- Dangling pointer and memory leakage.
- Destructor
- Multiple File Demo

constant example (demo01)

Static Data members (demo02 & demo03)

- we make the data members as static when we want to share the value across multiple objects.
- static data members must be initialized outside the class using classname and scope resolution operator.
- the static data members gets space on to the data section only once.
- all the objects access the same static data member from the data section, that means when any one object changes the value of such member, then the chages are reflected in all the remaning objects.

static Member Functions (demo02 & demo03)

- we can even make the member functions as static.
- static member functions are designed to call on class name using scope resolution operator.
- as they are designed to call on clas name this pointer is not passed to the static member functions.
- static member functions can only access static data members inside them. they cannot access non static data members.

Reference (demo04 and demo05)

- reference is an alias for an existing memory location.

Dynamic Memory Allocation (demo06 to demo08)

- If we want to allocate memory dynamically in CPP we have to use 'new' operator and to deallocate we have to use 'delete' operator
- if dynamically allocated memory is not deallocated/deleted them memory leakage happens.
- to avoid this leak its mandatory to deallocate the memory.

- if after deallocation we use the pointer to access the memory then our program may terminate or even we can get some garbage values.
- the pointer which points to the deallocated memory is called as dangling pointer.
- to avoid dangling pointer we need to assign it to NULL after deallocation.

use of destructor(demo09)

Multiple Files Demo(demo10)

- generally the class declarations and definations are seperated.
- class declarations are kept inside .h file.
- class defineations are kept inside .cpp file.
- to define the member functions outside the class we need to use class name and scope resolution operator.
- while declaring the header file always use header guard which helps us to avoid multiple inclusions.
- when user defined header file is included always include by using "" and .h extension.
- Header guard

```
# ifndef HEADERFILENAME_H
# define HEADERFILENAME_H

// keep the class declaration

# endif
```

Lab Work

Dynamic Memory Allocation