Agenda

```
Constructor
    Default/Paramaterless
    Parameterized
Destructor
Constant
    Data members
    Member functions
Constructor members initializer list
Reference
Copy Constructor
Dynamic memory allocation
```

Constructor (demo.cpp and demo01.cpp)

```
- job of a ctor is to initialize the data members of your class
- if user does not provide any ctor then compiler adds a ctor in your class which
is called as default ctor
- if you want to provide default values to your datamembers when object is created
then  write your own ctor.
- Ctor gets called only once in life time of an object
```

why ctor is special member function?

```
ctor is a special memebr function beacuse of three reasons as below
- Ctor name is same as that of class name
- It has no any return type
- It gets called automatically at the time of object creation
```

Types of Ctor()

```
- Default/parameterless Ctor
- Parameterized ctor
- Copy Ctor
```

Constant

```
1. Datamember (demo02.cpp)
    - If data memebr is constant then it needs to be initialized inside Ctor
```

```
members initializer list()
2. Member Function (demo03.cpp)
    - If you dont want to modify any non const datamember inside a member function then such member functions should be made as constant
    - Generally all your inspectors/print functions should be made as constant
```

Destructor (demo04.cpp)

```
- It is special member function similar to your constructor but with a tild sign(~)
- It gets called automatically when your object is aboout to go out of scope
- The dtor calling sequence is exactly opposite to that of ctor calling sequence
- If you alloacte dynamic memory to your data members then that memory should be released inside a dtor.
- If we have alloactaed any resourses then even those resources should be released inside dtor.
```

Reference (demo05.cpp and demo06.cpp)

```
- It is an alias given to the memory loaction
```

Copy Constructor (demo07.cpp)

```
- If you create an object by assingning it with already created object then your copy ctor gets called.
- If user does not provide a copy ctor the default copy ctor gets called.
- Copy ctor is a single parameterized ctor which takes reference of your class object as a argument
```

Dynamic memory allocation(demo08.cpp)

```
- to allocate memory dynamically we use new operator
- to deallocated the memrory we use delete operator
```

Homework

```
- Work with reference
- work with dynamic memory allocation
- revise the ctor and copy ctor demo.
- if you wish to learn the concept of shallow copy and deep copy then do the above
given homework.
```