# *DYNAMIC CONSTRUCTOR*

- When allocation of memory is done dynamically using dynamic memory allocator new in a constructor, it is known as dynamic constructor. By using this, we can dynamically initialize the objects.

# DESTRUCTOR

- A destructor, as the name implies is used to destroy the objects that have been created by a constructor.
- Like a constructor, the destructor is a member function whose name is the same as the class name but is preceded by a tilde. For Example:- ~ integer( ) { }
- A destructor never takes any argument nor does it return any value.
- It will be invoked implicitly by the compiler upon exit from the program to clean up storage that is no longer accessible.
- It is a good practice to declare destructor in a program since it releases memory space for future use.
- Delete is used to free memory which is created by new.
- Example:-

```
matrix : : ~ matrix( )
    {
        for(int i=0; i<11;i++)
        delete p[i];
        delete p;
    }
```

# UNION

- unions are the same as the structure type, means a collection of similar and different data types.
- But where the memory location of the members of the structure is different for each member, the union reserves same location in the memory to all its data member, due to which they affect each other,
- When you enter the second value after entering the first value, the second value erase the first value, in this way only the value entered in the last is stored, and when we print these values, in the place of erasing values, we get garbage values, that is only the value entered in last in print.
- In the diagram below, you can see that the union reserves the same location in the memory for its data members
- That is, the memory address of all the members is the same.