Chapter 6

- Class: collect of data & functions
- Constructor: with Arg.()

Ans(): data(0); without Arg()

Use to intializtion

- Protect: access in space of class
- Public :access in & out scope of class
- Destructor: ~Destructor_name();

(One of Destructorusing is deallocate memory)

- Form of Default copy constructor:

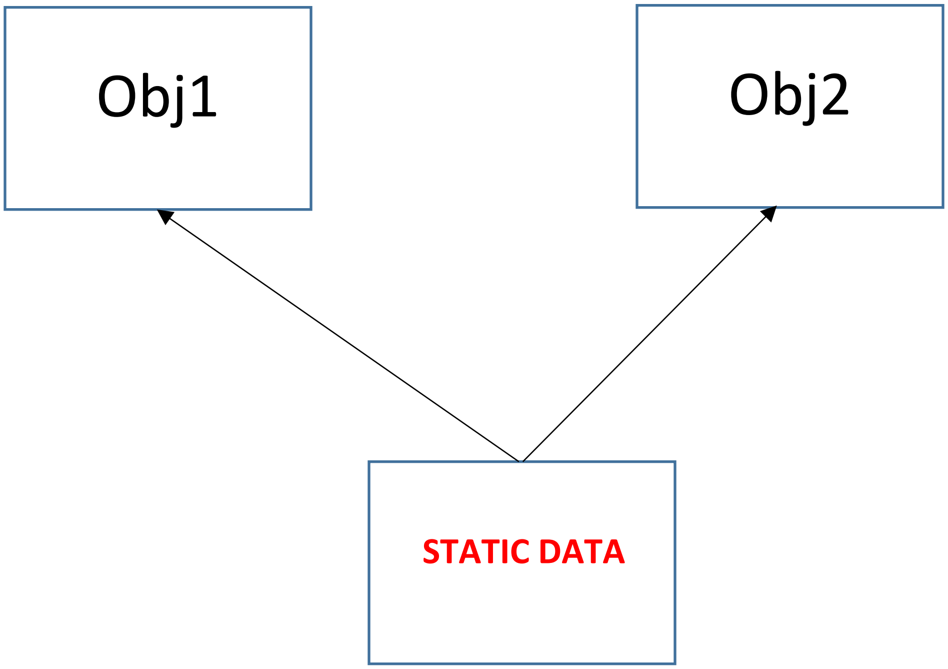
```
Main{
   classA=classB;
or
   classA(classB);
}
```

- Static data member :

Definination:intialize it global with value

Form: Datatype class_name::var_name =value;
Delecration: in private section without value

```
Ex. Class A{
Private:
        Static int var_static;
}
```



Used with All obj of class. Not initialize in construct As it will destroy main concept of static when call a new object.

- Declare function out of scope of class:
  Return name of name of ( ) { }
  Value class function

- Const .(write with declare & define function)
  Form: return name of ( ) const { }
  value func.
  This way Don't Allow to change data of this object in function.

- Const Argument member function
  Form: return name of (const data name);
  Value func. Type

- Const object :
  Form: const name of name of
  Class obj

  Can use only with const function AS we can't change data of this obj
  This obj can take it's value only with constructor

- Adivce: Try to increase const Function.

- Return class: name of name of (name of name of)
  Class func. Class Arg.

```
Class Base
{
  Base add(Base x){
  Return Base (data of Base)
  Or
  Base y;
  Return y;
  }
}
```

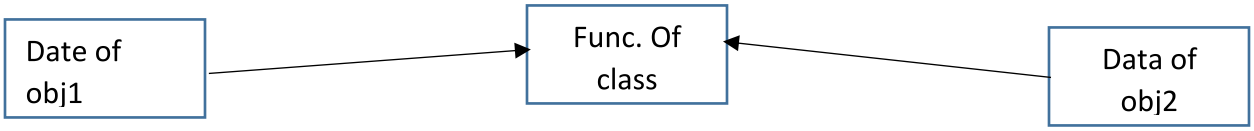
Conclude form that :

Return (obj of same class)

Or

Return (value of data of this class)

Note: All datas of objects deal with same function of class not copy of it


Date of obj1
Func. Of class
Data of obj2