

# MODULE-6

## C++

(POP)

Procedure oriented programming

(1) pop follows a top-down approach.

(2) pop → divide into small part → called as function

pop does not have any proper way for hiding data so  
(3) It is less secure

(4) It's deal with algorithm

(5) it need less memory

(6) There is no any access specifier

data are global. publicly  
Therefore less secure

(7) In pop, overloading is not possible

pop ek hi function hum  
pure koon nhi kar sakte  
hai

(8) ex:- C, Fortran, Pascal  
VB.

(OOP)

Object oriented programming

(1) oop follows a bottom-up approach

(2) oop → divide into part → called as object

(3) oops provide data hiding  
So it provides more security

(4) It's deal with data

(5) it need more memory

(6) There have access specifier

named → Public  
          → Private  
          → protected etc.  
data are public  
private  
Therefore more secure

(7) In oop, overloading is possible

✓ → function overloading  
✓ → operator overloading

oop ek hi function hum  
pure koon (kayid) kar sakte  
hai

(8) ex C++, Java, python etc  
VB.NET

less ~~data~~ Secure

more Secure

(i) In pop,

data  $\xrightarrow{\text{can move freely}}$  From  $\text{fun} \rightarrow \text{fun}$

in the system

(i) In oops,

data  $\xrightarrow{\text{cannot move easily}}$  From  $\text{Fun} \rightarrow \text{Fun}$

in the system

(ii) To add new

data  $\rightarrow$  function in pop  
is not so easy

(ii) To add new

data  $\rightarrow$  function in oops  
is so easy

(iii) In pops,

most function  $\rightarrow$  global data

For sharing that can be accessed  
freely from  $\text{fun} \rightarrow \text{fun}$   
in the system

(iii) In oop,

data cannot move easily  
from  $\text{fun} \rightarrow \text{fun}$

It can be kept  $\rightarrow$  data public  
or private

So we can control the access  
of data



① - is a special member function which name is the same as the name of its class in which it is declared and defined.  
 ② - purpose of constructor is to initialize the object of the class.

### Constructor

(1) It allocates memory to an object.

(2) In a class, there can be multiple constructors.

(3) It is declared

className (argument if any)  
 { constructor's Body }

(4) Constructor

Can either accept argument or not

(5) Constructor

Can be overloaded

(6) constructor automatically called when object is created

### Destructors

(1) It deallocates memory to an object.

(2) In a class, there can be single destructors.

(3) It is declared

~className (no arguments)  
 { }.

(4) destructors

it can't have any arguments

(5) destructure

cannot be overloaded

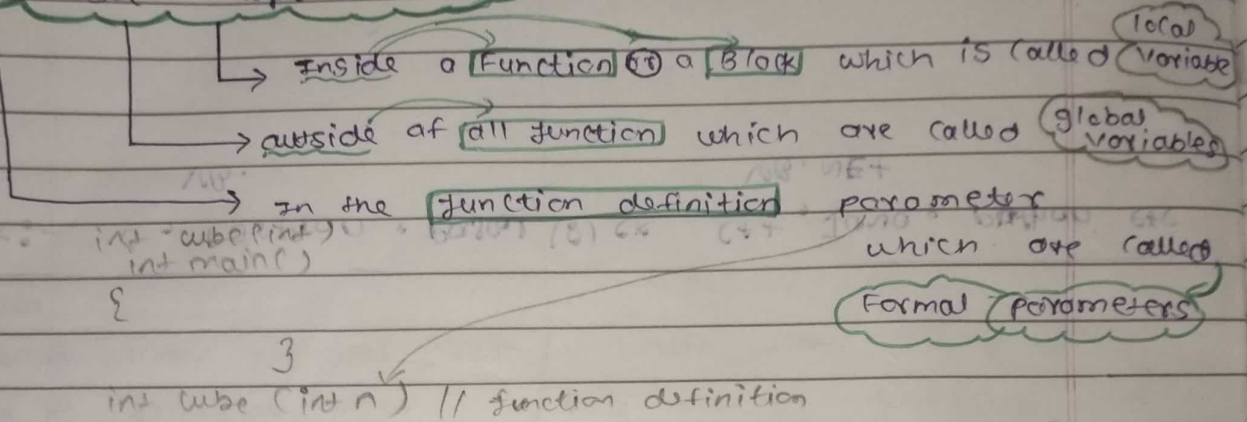
(6) destructor automatically called when object terminate

Class (if dequ)

?

}

## Variable Scope in C++



## The iostream file :

#include <iostream>

for Dev++  
codeBlock

#include <iostream.h>

for Turbo C

- Istream class supports → **Input Function**

- Ostream class supports → **output Function**

- All objects are included in **iostream** file so, we can use **input** or **output** statement in our program.



Access specifier : // Can be private, public or protected

data members; // variable to be used

member Function() { } // method to access data function

### access specifier in C++

- Access specifier is a keyword

used to Control visibility or accessibility of class member (data member, member function)

Public :- public member are accessible from anywhere in the program.

- They can be accessed by Object of the class, as well as by external function or other classes

Private :- private member are accessible with in the class & self.

- They cannot be accessed by Object of class, as well as by external function or other classes

protected :-

protected member are similar to private member in that ~~they are not~~ they are not directly accessible from outside the class.

syntax :-

```
class Classname
```

```
{  
    public :
```

```
        Data member ;
```

```
        member function ;
```

```
    private :
```

```
        data member ;
```

```
        member function ;
```

```
    Protected :
```

```
        Data member ;
```

```
        member function ;
```

```
}
```

```
ex:- #include <csdio.h>
```

```
using namespace std;
```

```
class Class3
```

```
{
```

```
    Private :
```

```
        int a;
```

```
    Public :
```

```
        display()
```

```
{
```

```
    cout << "display function";
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    class3 ob;
```

```
    ob.a = 10;
```

```
    ob.display();
```

```
    return 0;
```

```
}
```

(can be access through  
other function call  
in class)

```
Private :
```

```
    int a;
```

```
public :
```

```
    display (int b)
```

```
{  
    a = b;
```

```
    cout << "value of a = " << a;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    class3 ob;
```

```
    ob.display (10);
```

```
    return 0;
```

```
}
```

value of a = 10

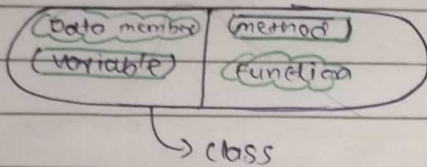
give - error

in private specifier  
directly does not access  
the member

## # Encapsulation :-

- The wrapping up of data member and function into single unit (called class)

Encapsulation in C++



is known as encapsulation

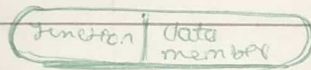
Encapsulation Representation

- If we all data member are <sup>put</sup> in private there are fully encapsulated class

e.g. Private :

```
int num;  
string label;
```

- therefore data is not accessible to the outside world
- only those function which are wrapped in the class can access it.



- This insulated of data from direct access by the program

is called data hiding or information hiding

advantage :- ~~For data hiding & increase security~~

(1) ~~Increase code Res~~

Advantage :-

- In encapsulation, we can hide the data, which is better for security
- Increase code reusability.



## # Abstraction :

- data abstraction is one of most essential and important feature of oops in c++
- Abstraction mean display → only essential information  
↓  
hiding the  
without include background details
- Control abstraction through access specifier
  - (1) member declared as public in a class can be accessed from anywhere in the program
  - (2) member declared as Private in a class can be accessed only from within the class
  - (3) member declared as Protected in a class are a special kind of access specifier
    - It works similarly to private and can access it

## Advantage :

- (1) In Abstraction, we can not allowing anyone else to see the background details, which is better for security.
- (2) Increase code reusability



# Inheritance :

- Inheritance is one of fundamental concept in C++

- It is allow to you create

(derived class)  
a new class

base on

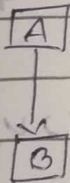
(base class)  
existing class

- new class / derived class → inherit the Properties and behaviour of the Base class

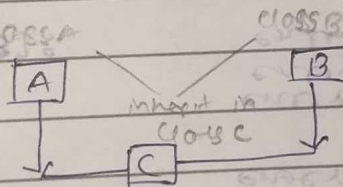
- Can also add it own unique properties and behaviours in derived class

- They are Five type of inheritance

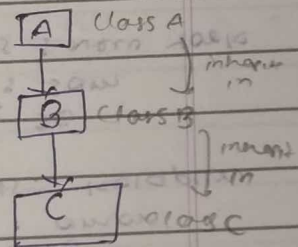
Single inheritance



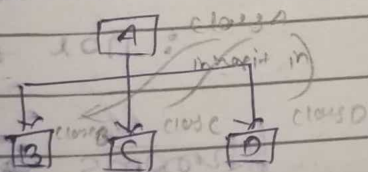
multiple inheritance



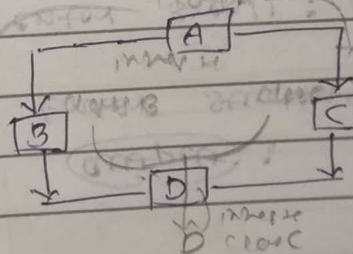
multilevel inheritance



hierarchical inheritance



hybrid inheritance



Type of inheritance