

* Difference between Class and Object.

Object

1) Object is an instance of a class.

2) Object is a physical entity.

3) Object is created many times as per requirement.

4) It allocates memory space when created.

5) Objects can be manipulated.

6) It does not have any values which are associated with fields.

7) ~~To~~ To create an object in C++, we've to specify the class name, followed by object name.

Class

Class is a blueprint from which objects are created.

Class is a logical entity.

Class is declared once.

It does not allocate memory space when created.

They cannot be, as they're not available in memory.

Each & every object has its own values, which are associated with the fields.

Class in C++ can be created using keyword 'class' with name of the class.

Example of Class :-

```

class Room {
public:
    double length;
    double breadth;
    double height;

    double calculateArea() {
        return length * breadth;
    }

    double calculateVolume() {
        return length * breadth * height;
    }
};

```

Example of Object :-

Room student;

student.length = 42.5;

student.breadth = 30.8;

student.height = 19.2;

cout << "Area of Room = " << student.calculateArea() << endl;

cout << "Area of Room = " << student.calculateVolume() << endl;

return 0;

}

C++ Inheritance →

The capability of a class to derive properties and characteristics from another class is called Inheritance.

Includes:-

Base Class / Parent Class → The class whose properties are inherited by sub class / child class.

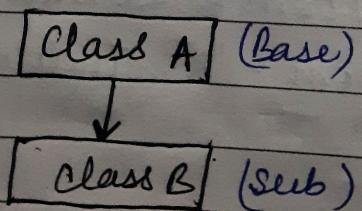
Sub Class / Child class → The class that inherits properties from another class or Base / parent class.

Types Of Inheritance →

1) Single Inheritance -

In this, a class is allowed to inherit from only one class.

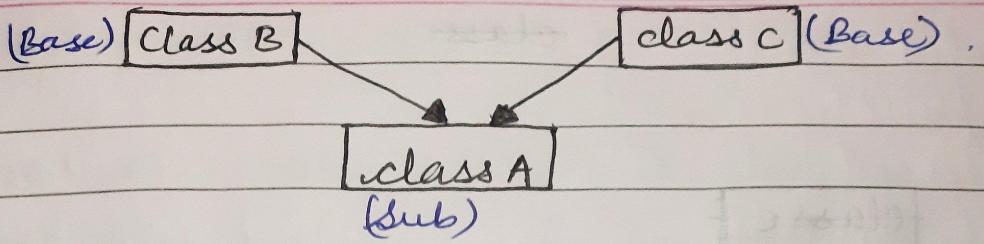
One sub class is inherited by only one Base class.



2) Multiple Inheritance -

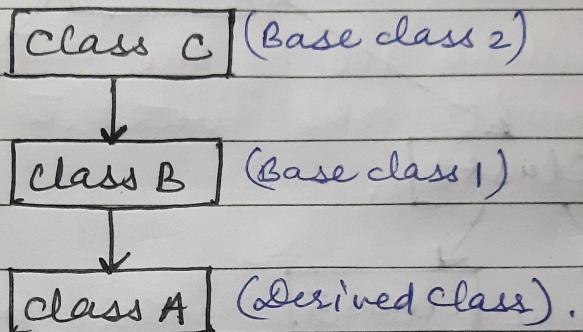
In this, a class can inherit from more than one class.

One class is derived from more than one Base class.



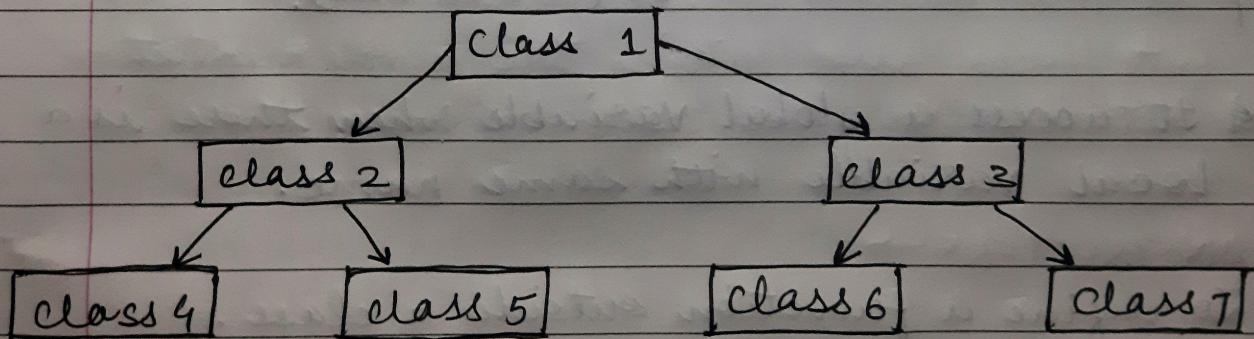
3) Multilevel Inheritance —

In this, a sub-class/derived class is created from another derived class.



4) Hierarchical Inheritance —

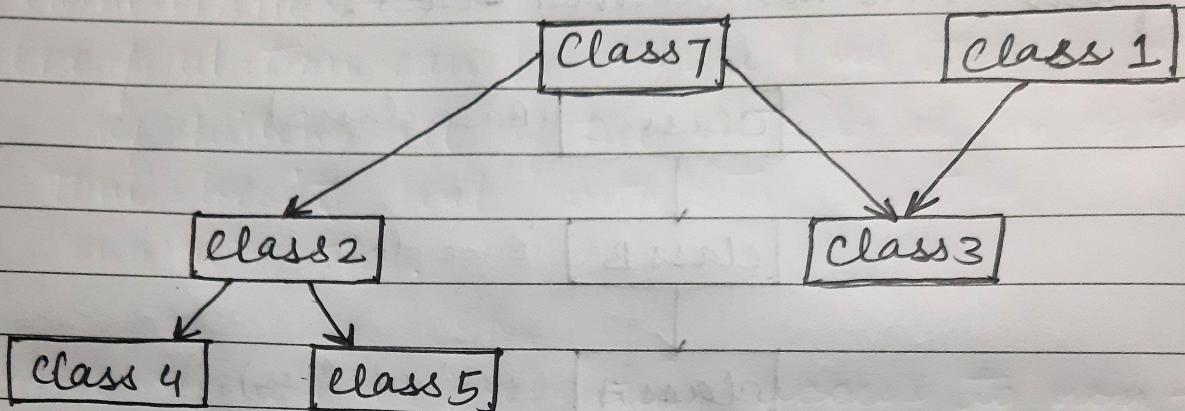
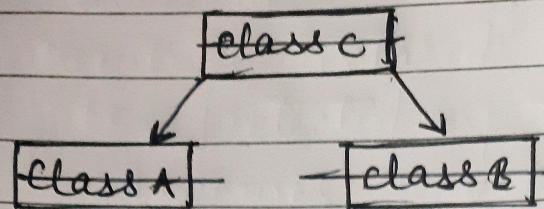
In this, more than one sub/derived class is inherited from a single base class.



5) Hybrid Inheritance —

In this, more than one type of inheritance is combined.

For e.g., Hierarchical & Multiple Inheritance.



SCOPE RESOLUTION →

Operator is shown as :: (two colons).

- To access a global variable when there is a local variable with same name.
- To define a function outside a class.
- To access a class's static variable.
- To differentiate between same variable name existing in two previously made classes, we use Scope Resolution (for multiple Inheritance).

- Used for Namespace.
- Used when we refer to a class inside another class.

Access Modifier →

Access Modifiers are the techniques that are used / applied to members of class to restrict their access beyond the class.

These modifiers can be achieved by using the three keywords →

- 1) Public
- 2) Private
- 3) Protected

- Public modifier is available to all, or it gives access to all. All members of the class will be available to everyone after declaring them as public.
It can be accessed anywhere outside the class but within the program.
We use (.) dot operator to directly access member functions & data.
- Private member keeps the data members inside the class, so the function inside the class can access class members declared as private. You can't access members outside directly by any object or function from outside the class.

- Protected modifier keeps the data members or functions safe and they can't be accessed directly from other classes. Members declared in 'protected' can only be protected up to next level then it becomes private.

★ ENCAPSULATION →

Encapsulation can be defined as the process of hiding all of the details of an object that doesn't deal with its essential characteristics.

OR

Also defined as preventing access to non-essential details/info. of classes or its objects.

★ ABSTRACTION →

It is the mechanism of exposing only the interfaces and hiding the implementations details from the user.

★ CONSTRUCTORS →

- Special type of 'member function' that is called automatically when an object is created.
- ^{has} Same name as class. Does not have return type.

Types:-

- 1) Default
- 2) Parameterized
- 3) Copy

Default has no parameters. ()

Parameterized includes parameters.

Copy used to copy data of one object to another.

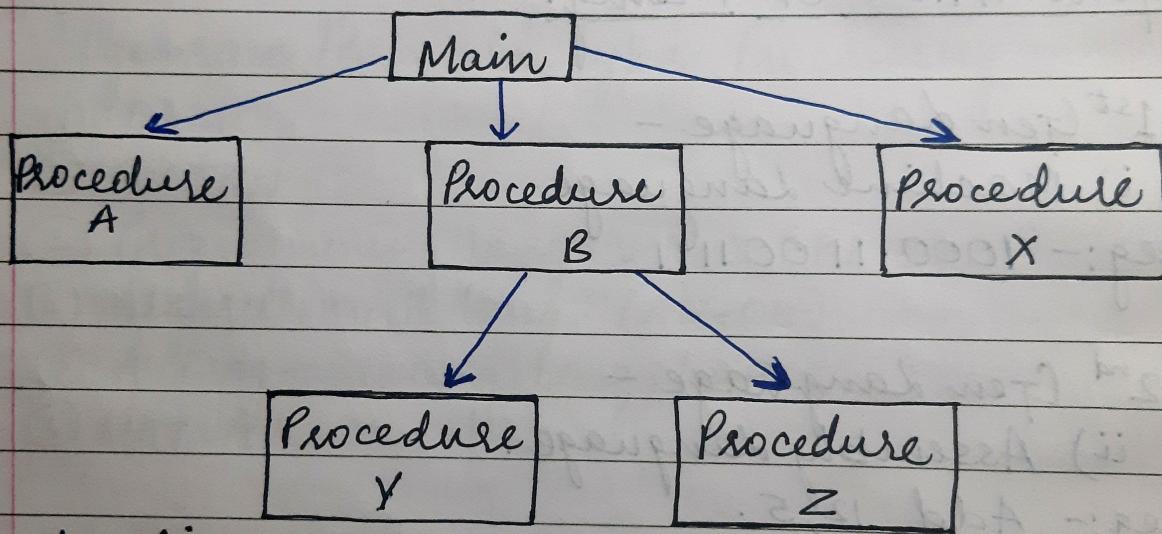
* DESTRUCTORS →

- It destructs the objects of classes.
- can be defined only once in a class.
- It is also invoked automatically.
- Prefixed with tilde (~) sign, it is defined like constructor & must have same name as class.

Destructor is called when the object goes out of scope:-

- when the function ends
- the program ends
- a block containing local variable ends.
- a delete operator is called.

★ Procedure Oriented Programs (POP) →
 consists of writing a set list of instructions
 for a computer to follow and organise
 These instructions into groups known as
 functions. It is a top down approach.



Functions →

- ① Emphasis is on algorithm
- ② Large programs are divided into smaller ones called functions.
- ③ Most of the functions share global data.
- ④ Data moves around anywhere from function to function.
- ⑤ Function transforms data from one form to another.
- ⑥ It employs top-down approach for program designing.

Advantages →

- ① Easy to use & understand.
- ② Easy to learn.
- ③ Error can be easily removed.
- ④ Simple programming.

Disadvantages →

- ① It is time consuming.
- ② No security ~~for~~ of data.
- ③ No structure or code reusability.
- ④ Slow performance.
- ⑤ Difficult for maintenance.
- ⑥ There's no proper exception handling.

✓ * Object Oriented Programming (OOPs) →

In OOPs Languages all the applications are built around class and objects.

Features →

- ① Emphasis is on data but no procedure.
- ② Programs are divided into objects.
- ③ Data structures are designed such that they characterise the object.
- ④ Functions that operate on data of an object are tied together in the data structure.
- ⑤ Data is hidden and cannot be ~~be~~ accessed by external functions.
- ⑥ New data & functions can be easily added whenever necessary.
- ⑦ It follows bottom-up approach in prog. designing.

2/12/2021

Date:

2/1

Advantages →

- ① Improved reliability & flexibility.
- ② Real world modelling.
- ③ Modifiability.
- ④ Extensibility.
- ⑤ Reduced maintenance.
- ⑥ High code reusability.

Disadvantages →

- ① OOP development is not a solution.
- ② It is not a techo technology.
- ③ It is not yet completely accepted by major vendors.

OOP Concepts →

- ① Objects → Objects are the basic run-time entities in object oriented system with some characteristics & behaviours.
- ② Classes → A class is a collection of objects of similar types, that share common property & relationship.
- ③ Data abstraction → Abstraction refers to the act of representing essential features without including the background details & explanation.
- ④ Encapsulation → Wrapping up of data or methods in a class is known as encapsulation.

2/12/21

Page No. :

Date :

① Inheritance → The mechanism of deriving a new class from a base class is called inheritance.

② Poly morphism → It is the concept that supports the capability of an object of a class to behave differently in response to a message action.

इसमें सभी धैर्यिक नहीं लिखे हैं; कृपया वे धैर्यिक स्वयं ही पढ़लें।

