Downloaded from www.studiestoday.com

Unit-I
Object Oriented Programming in C++

Inheritance Chapter: 04

Inheritance is the process of creating a new class from existing class/classes. The existing class is known as the base/super/parent class and newly created class is known as derived/sub/child class. The derived class will inherits the properties of base class.

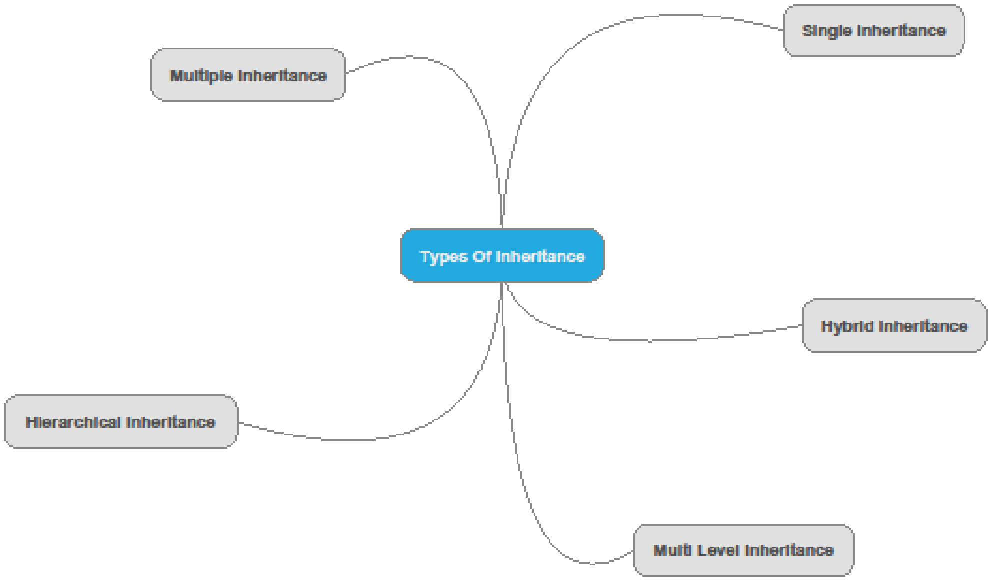
Advantages of Inheritance are given below:

Reusability: It helps the code to be reused in derived class. The base class is defined and once it is compiled, it needs not to be reworked.

Transitivity: If class B inherits properties of another class A, then all subclasses of class B will automatically inherits the properties of A. It is called transitive property.

Types of Inheritance:

1. **Single inheritance:-** When a sub class inherits only from one base class, is known as single inheritance.
2. **Multiple Inheritance**:- When a sub class inherits from multiple base classes, is known as multiple inheritance**.**
3. **Hierarchical Inheritance**:- When many sub classes inherit from a single class, it is known as hierarchical inheritance.
4. **Multilevel Inheritance:-**When a class inherit from a class that itself inherits from another class it is known as a multilevel inheritance.
5. **Hybrid Inheritance:** It is a combination of 2 or more of above types of inheritance. There is no pattern of deriving from classes.



Syntax for defining a derived class:

class<derived class name>**:**<visibility mode><base class name>

{

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

```
//Data members of derived class
//member functions of derived class
};
```

Visibility modes

The visibility mode in the definition of the derived class specifies whether the features of the base class are privately derived or publicly derived or protected derived.

Visibility Mode	**Public member of base class becomes**	**Protected member of base class becomes**	
Public	Public in derived class	Protected in derived class	Private member of base class are not directly accessible to derived class
Protected	Protected in derived class	Protected in derived class	
Private	Private in derived class	Private in derived class	

Constructor and Destructor in Derived classes:

When a base class and a derived class both have constructor and destructor , the constructors are executed in order of inheritance and destructors are executed in reverse order. That is, the base constructor is executed before the constructor of the derived class and the destructor of the derived class is executed before the base class destructor.

Short Anwer Type Questions(2 Marks)

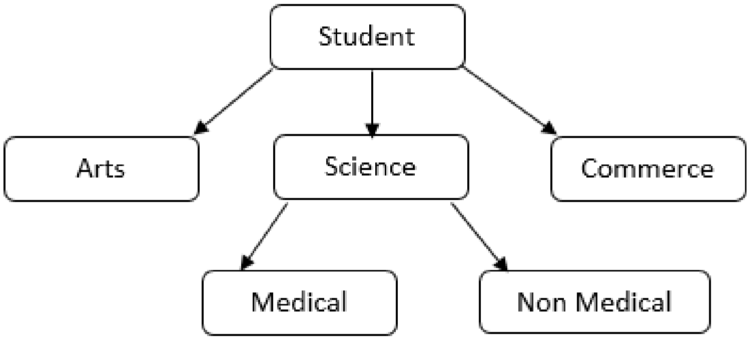
Q1.What do you understand by Inheritance?

Q2.What is Base class and Derived class?

Q3.What is difference between Multiple Inheritance and Multilevel Inheritance?

Q4.What is difference between public access specifier and private access specifier?

Q5.Which type of Inheritance is depicted below?



Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

Q6.Consider the following code and answer the questions below:

```
#include<iostream.h>
class employee
{       int emp_no;
        char emp_name[25], emp_addr[30], emp_dept[10];
        public:
                void emp_input();
                void emp_print();
};
class manager: public employee
{       int no_of_emp;
        public:
                void input();
                void print();
};
```

i. Which type of inheritance is shown in the above code?
ii. Name the base class and the derived class.
iii. Name all the private data members of class employee.
iv. Name all public member functions of class manager.

Downloaded from www.studiestoday.com