

**Practical Lecture : Type  
conversion**



# Quick Recap

Let's take a quick recap of previous lecture –

- Basic concept of type conversion
- Type conversion- implicit and explicit
- Difference between implicit and explicit conversion
- Basic type to class type
- Class type to basic type
- One class to another class type

# Today's Agenda

Today we are going to cover -

- Inheritance basics – base class , dervied class
- Type of inheritance- simple, multi-level, multiple and hierarchical
- Access specifier or mode (private, protected, public inheritance)

**Let's Get Started-**

# Inheritance basics

Inheritance is one of the object oriented programming paradigm as mentioned initially

Inheritance is the process of using properties of one class into the another class

This is achieved by deriving sub-class from the base class.

A class that is inherited is called a super class, base class or parent class and the derived class is called a sub-class, derived class or child class.

A sub-class is a specialized version of a super class.

Eg. we can categories the 'animal' into two categories: 'wild animal' and 'pet animal'. Also we can categories 'wild animal' into 'tiger', 'lion', 'leopard' and 'pet animal' into 'cat', 'dog', 'bull'.

# Inheritance basics

It inherits all of the instance variables and methods defined by the super class and add its own, unique elements.

Inheritance provides the facility of re-usability.

We can add new features (new data and function) into existing class without modifying it.

This is done by deriving new class (subclass or child class) from existing class (super class or parent class).

The sub class contains the facility of super class as well as its own features..

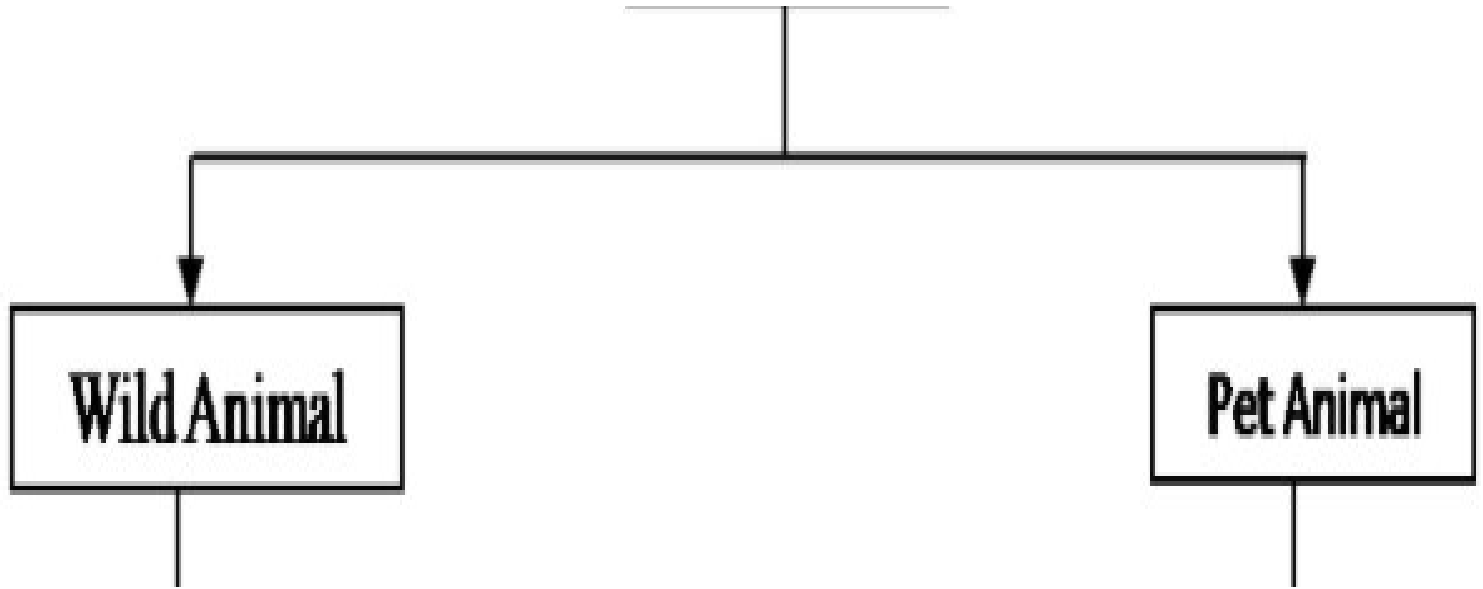
# Advantages of inheritance

- Application development time is less.
- Application take less memory.
- Application execution time is less.
- Redundancy (repetition) of the code is reduced or minimized so that we get consistence results and less storage cost.

## Explanation:

Inheritance provides the facility of re-usability. Means Instead of writing the same code, again and again, we can simply inherit the properties of one class into the other. This makes it easier to create and maintain an application. OOP is all about real-world objects and inheritance is a way of representing real-world relationships.

## Knowledge check- question

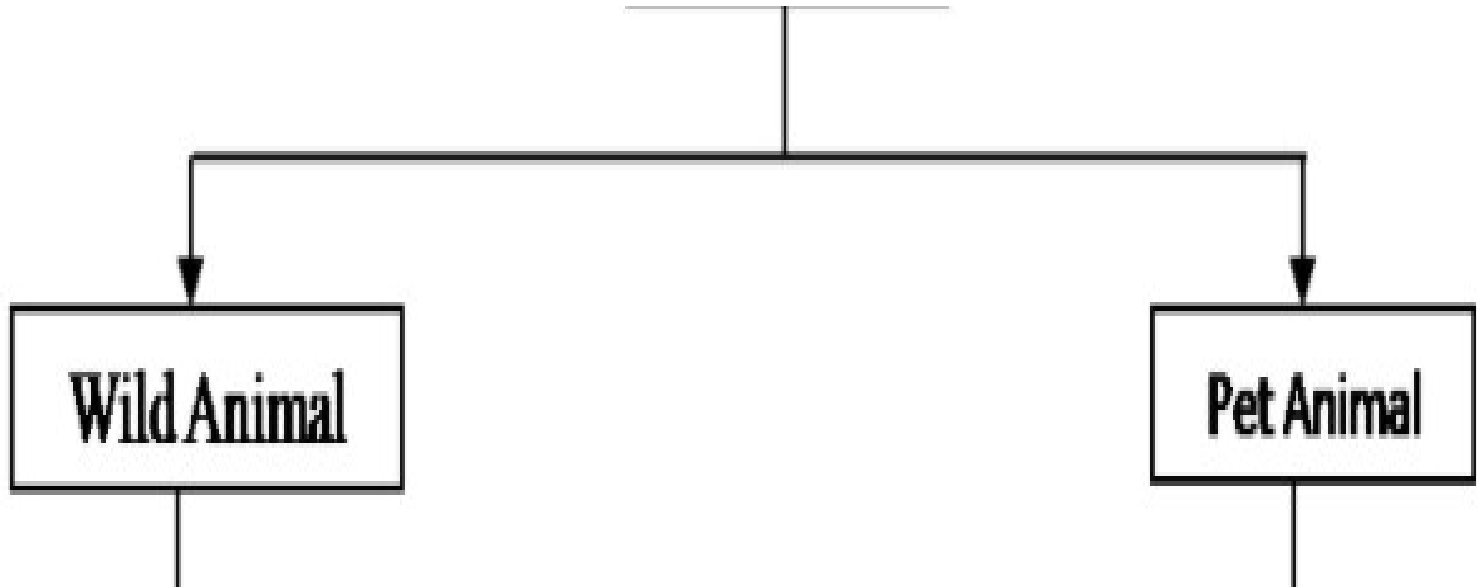


Identify the base class and derived classes in the above figure.  
What is 'Animal' class called here? What about rest all classes

Type the answers in the chat box.



## Knowledge check - Answer



1. Base class : Animal and derived classes : Wild animal and Pet animal;  
Base class : Wild animal , child classes : Tiger, Lion, Leopard;  
Base class : Pet Animal , child classes: Cat, Dog, Bull
2. 'Animal' class : Base class /super class / Parent class  
Wild Animal, Pet Animal: Derived class/ Sub class/ Child class.

# Types of inheritance

1. Single inheritance : This is a form of inheritance in which a class inherits only one parent class.
2. Multi-level inheritance : In this form of inheritance , a base class is inherited by a derived class, which further becomes base class and inherited by next level derived class and so on
3. Multiple inheritance : Here a class inherits more than one parent class.
4. Hierarchical inheritance: In this, various child classes inherit a single Parent class.
5. Hybrid inheritance: It is the combination of multi-level, multiple and hierarchical inheritance.

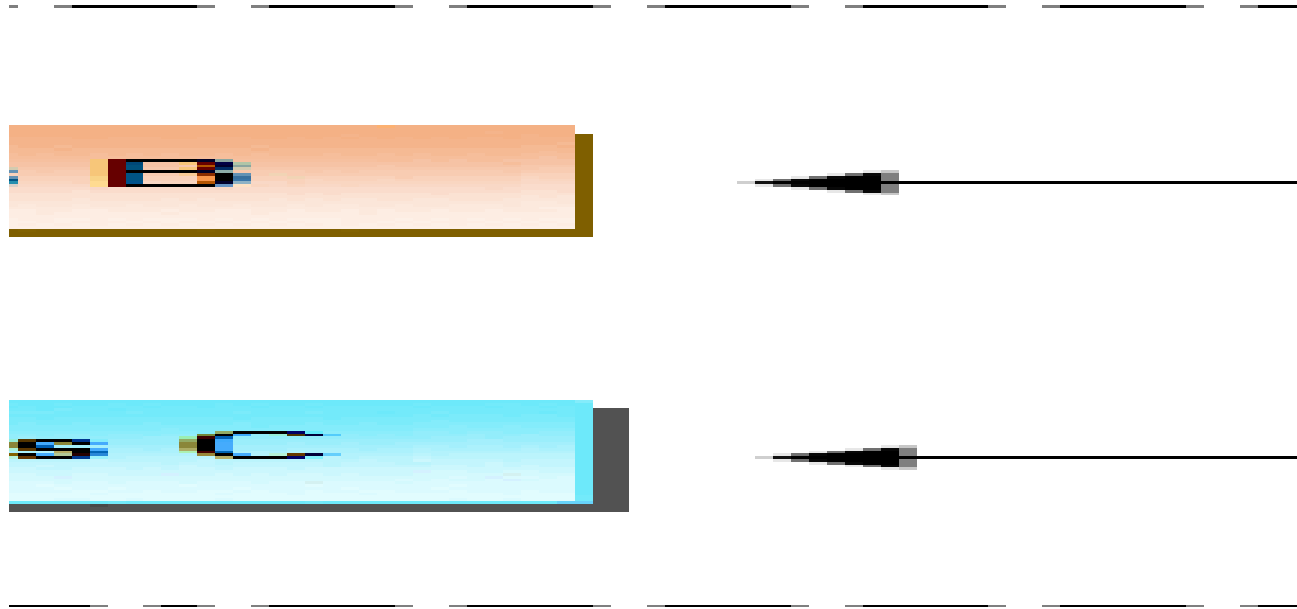
**Note: <Please encourage students to give various real –life examples of all above classes>**

# Single Inheritance



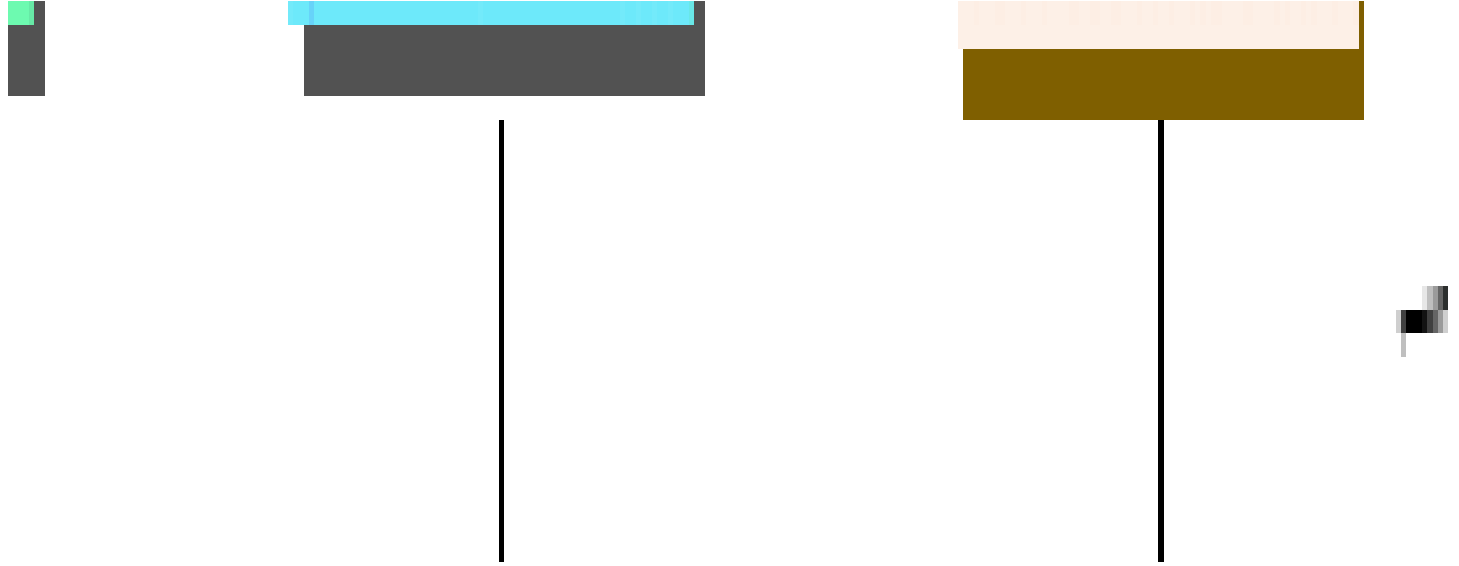
Eg. Parent-child, Animal- Dog, Fruit - Apple , doctor- pediatrician

# Multi-level Inheritance



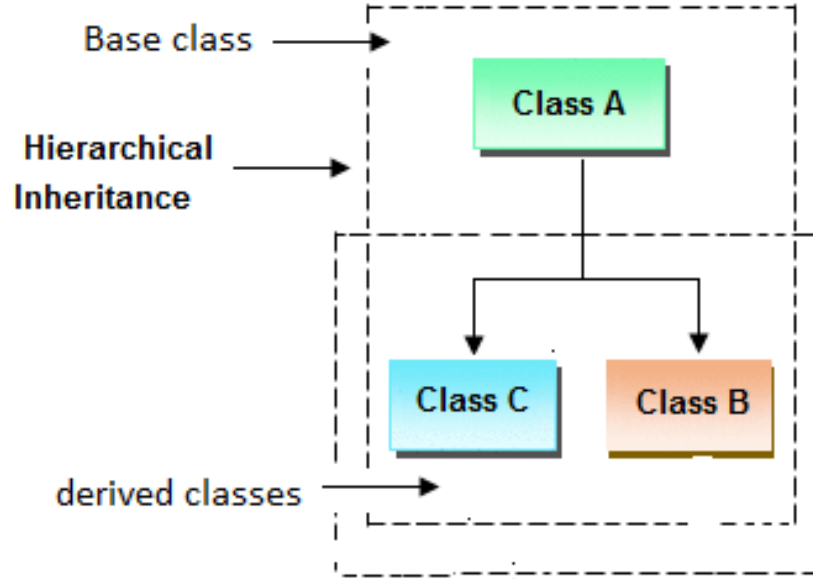
Eg. Grandfather- Father- Child, Vehicle-Car- Audi, Doctor- Orthopedic- KneeSurgeon

# Multiple Inheritance



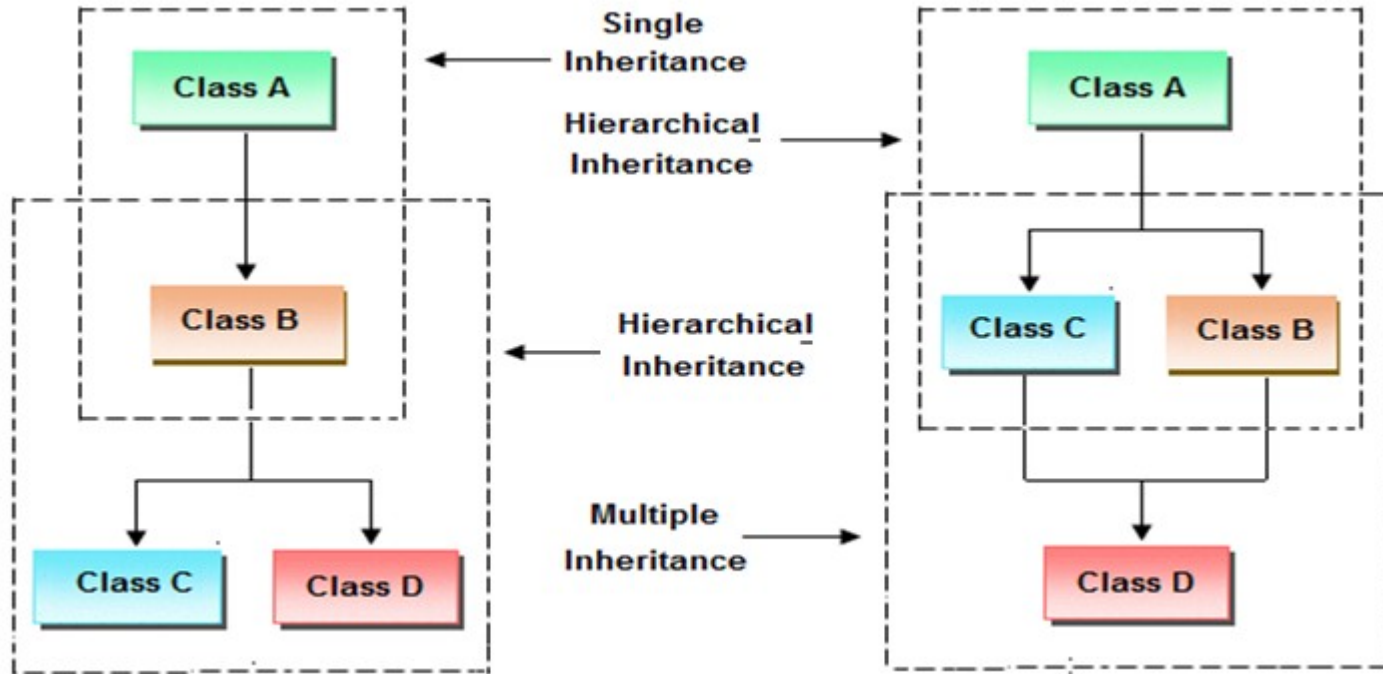
Eg. Mother, Father- Child , student , Teacher- Teaching Assistant

# Hierarchical Inheritance



Eg. Animal- Dog, lion, cat etc,  
Fruit- Apple, Mango etc,  
Person- student ,Teacher, scientist, Engineer etc

# Hybrid Inheritance



Eg: Person- student ,Teacher – Teaching Assistant

# Syntax of class derivation

```
class BaseClass{  
    // members....  
    // member function  
}
```

```
class DerivedClass : public BaseClass{  
    // members....  
    // member function  
}
```



# Syntax of class derivation

A class can be derived from more than one classes, which means it can inherit data and functions from multiple base classes.

To define a derived class, we use a class derivation list to specify the base class(es).

A class derivation list names one or more base classes and has the form –

***class derived-class: access-specifier base-class***

Where access-specifier is one of public, protected, or private, and base-class is the name of a previously defined class.

If the access-specifier is not used, then it is private by default.

## Practice Question

Define a class student and a derived class test which stores marks of subject 1 and subject 2.

```
class student
```

```
{
```

```
    int roll_number;
```

```
public:
```

```
    void get_number(int);
```

```
    void put_number(void);
```

```
};
```

```
void student::get_number(int a)
```

```
{
```

```
    roll_number=a;
```

```
}
```

```
void student::put_number()
```

```
{
```

```
cout<<"Roll number= " <<roll_number<<endl;
```

```
}
```

# Practice Question

```
class test:public student
{
    float sub1;
    float sub2;
public:
    void get_marks(float,float);
    void put_marks(void);
};
void test::get_marks(float x,float y)
{
    sub1=x;
    sub2=y;
}
```

## Practice Question

```
void test::put_marks()
{
    cout<<"Marks in sub1="<<sub1<<endl;
    cout<<"Marks in sub2="<<sub2<<endl;
}
int main()
{
    test t1;
    t1.get_number(11);
    t1.put_number();
    t1.get_marks(75.0,59.5);
    t1.put_marks();
    return 0;
}
```

# Practice Question

Output:

Roll number= 11

Marks in sub1=75

Marks in sub2=59.5

# Assignment

Create a class Employee which stores and displays attributes of an employee like empname, empno, department, salary. Create a derived class called Project which allows to store project name. Write a C++ program to create object of project class.

Any Questions ??  
**Any Questions??**

# Thank You!

**See you guys in next class.**