Inheritance

Inheritance-

Inheritance is one of the key features of Object-oriented programming in C++. It allows us to create a new class (derived class) from an existing class (base class).

The derived class inherits the features from the base class and can have additional features of its own.

Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time.

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the base class, and the new class is referred to as the derived class.

Syntax:

```
class parent_class {
    //Body of parent class
};
class child_class: access_modifier parent_class {
    //Body of child class
};
```

Here, child_class is the name of the subclass, access_mode is the mode in which you want to inherit this sub-class, for example, public, private, etc., and parent_class is the name of the superclass from which you want to inherit the subclass.

Modes of Inheritance

1. **Public mode:** If we derive a subclass from a public base class. Then, the base class's public members will become public in the derived class, and protected class members will become protected in the derived class.

2. **Protected mode:** If we derive a subclass from a Protected base class. Then both public members and protected members of the base class will become protected in the derived class.

3. **Private mode:** If we derive a subclass from a Private base class. Then both public members and protected members of the base class will become Private in the derived class.

Base Class member Access Specifier	Public	Protected	Private
Public	Public	Protected	Private
Protected	Protected	Protected	Private
Private	Not Accessible	Not Accessible	Not Accessible

Example:
Suppose we have three classes with names: car, bicycle, and truck. The properties for each are as follows:

Car	Bicycle	Truck
➔ Colour ➔ Max Speed ➔ Number of Gears	➔ Colour ➔ Max Speed ➔ Is foldable?	➔ Colour ➔ Max Speed ➔ Max weight

From above, we can see that two of the properties: Colour and MaxSpeed, are the same for every object. Hence, we can combine all these in one parent class and make three classes their subclass. This property is called Inheritance.

Technically, inheritance is defined as the process of acquiring the features and behaviors of a class by another class. Here, the class that contains these members is called the base class, and the class that inherits these members from the base class is called the derived class of that base class.

Code of the above example:

```cpp
class vechile{
    public:
    string color;
    int max_speed;
};

class car : public vehicle{
    int num_gears;
};

class bicycle : public vehicle{
    bool is_foldable
};

class truck : public vehicle{
    int max_weight;
};
```