Subject Name: Object Oriented Programming Using C++

Subject Code: BCA-301 N

Subject Topic: Types of Inheritance and Members access specifiers

Abhishek Dwivedi

Assistant Professor

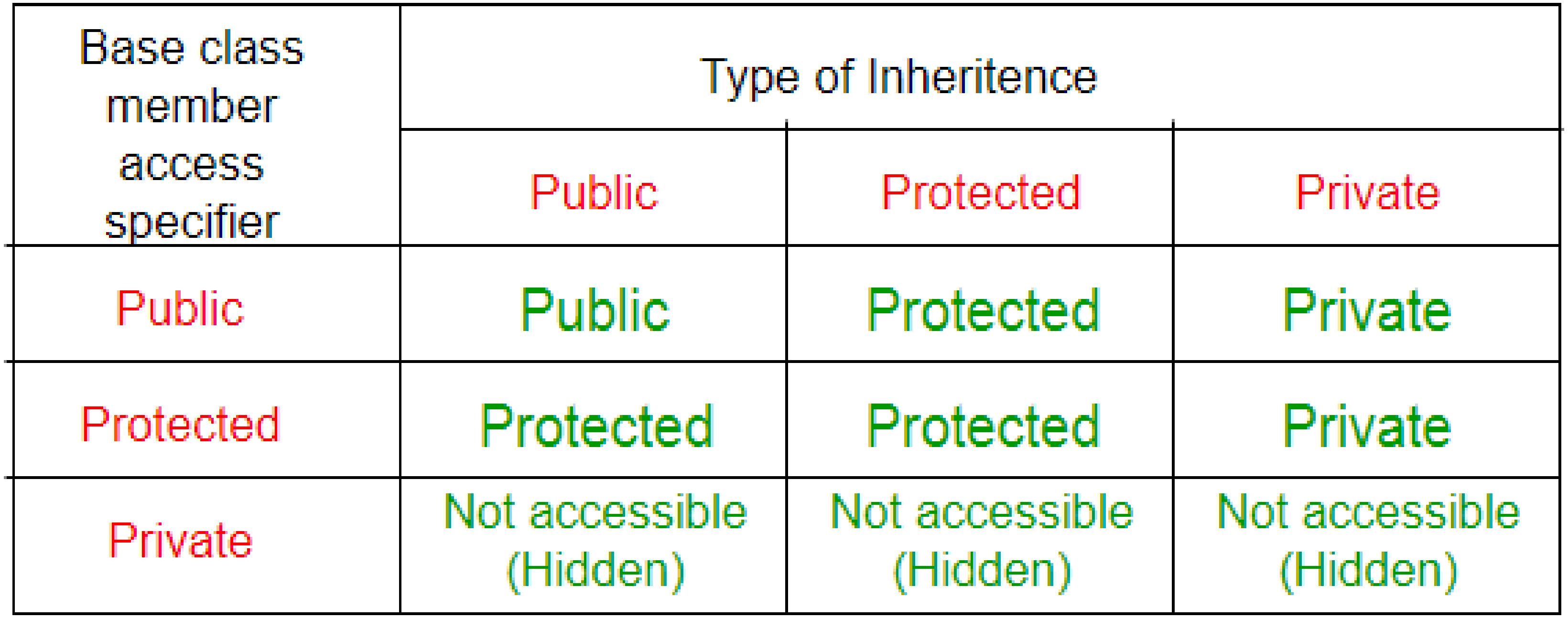
Department of Computer Application

UIET, CSJM University, Kanpur

Modes of Inheritance

- **Public mode**: If we derive a sub class from a public base class. Then the public member of the base class will become public in the derived class and protected members of the base class will become protected in derived class.
- **Protected mode**: If we derive a sub class from a Protected base class. Then both public member and protected members of the base class will become protected in derived class.
- **Private mode**: If we derive a sub class from a Private base class. Then both public member and protected members of the base class will become Private in derived class.

- The below table summarizes the above three modes and shows the access specifiers of the members of base class in the sub class when derived in public, protected and private modes:



Base class member access specifier	Type of Inheritence		
	Public	Protected	Private
Public	Public	Protected	Private
Protected	Protected	Protected	Private
Private	Not accessible (Hidden)	Not accessible (Hidden)	Not accessible (Hidden)

```
class A
{
public:
    int x;
protected:
    int y;
private:
    int z;
};

class B : public A
{
    // x is public
    // y is protected
    // z is not accessible from B
};

class C : protected A
{
    // x is protected
    // y is protected
    // z is not accessible from C
};

class D : private A    // 'private' is default for classes
{
    // x is private
    // y is private
    // z is not accessible from D
};
```

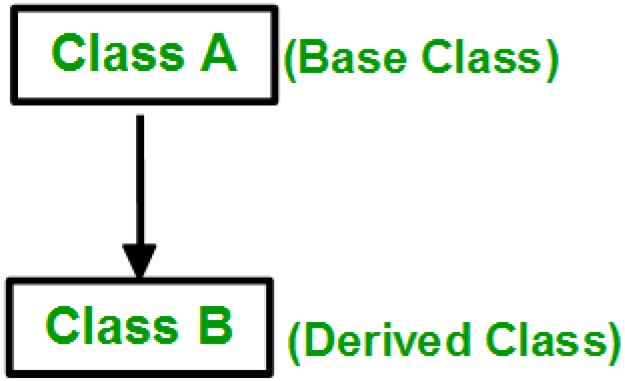
Types of Inheritance in C++

We have 5 different types of Inheritance. Namely,

- Single Inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Multilevel Inheritance
- Hybrid Inheritance (also known as Virtual Inheritance)

Single Inheritance in C++

- In this type of inheritance one derived class inherits from only one base class. It is the most simplest form of Inheritance.



Example

```
class Vehicle
{
  public:
    Vehicle()
    {
      cout << "This is a Vehicle" << endl;
    }
};

class Car: public Vehicle      // sub class derived from one base class
{


};

void main()
{
    // creating object of sub class will invoke the constructor of base classes

    Car obj;
    getch();
}
```

Another Example

```
class base                    //single base class
{
   public:
     int x;
     void getdata()
    {
         cout << "Enter the value of x = ";
         cin >> x;
    }
 };
```

```
class derive : public base              //single derived class
{
    private:
        int y;
    public:
        void readdata()
        {
           cout << "Enter the value of y = ";
           cin >> y;
        }
        void product()
        {
          cout << "Product = " << x * y;
        }
};
```

```
void main()
{
derive a; //object of derived class
a.getdata();
a.readdata();
 a.product();
getch();
}
```

References:

- www.studytonight.com
- www.tutorialpoint.com
- www.geeksforgeeks.org
- “Object oriented programming in C++” Robert Lafore
- “Object oriented programming with C++”, E.Balagurusamy