

19CSE201 :Advanced Programming

Lecture 10

More on Inheritance in C++

By

Ritwik M

Assistant Professor(SrGr)

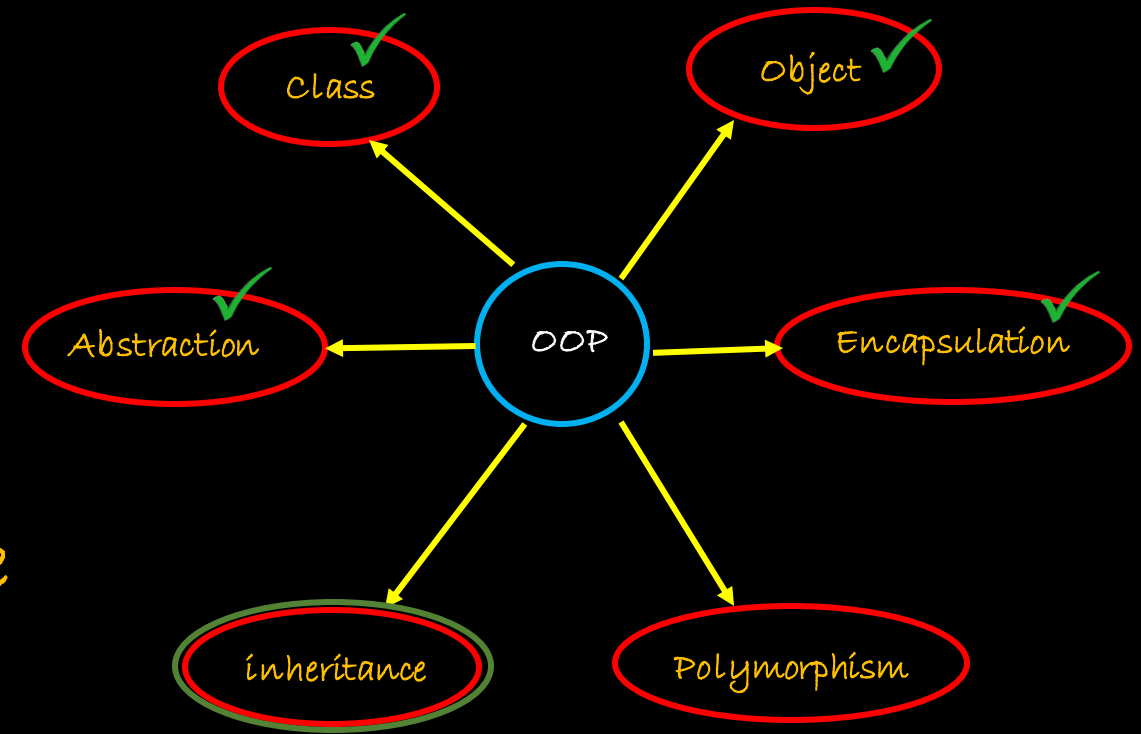
Dept. Of Computer Science & Engg

Amrita Vishwa Vidyapeetham -

Coimbatore

A Quick Recap

- Inheritance
- Why Inheritance
- Single inheritance
- Access specifiers and inheritance
- Examples and Exercises

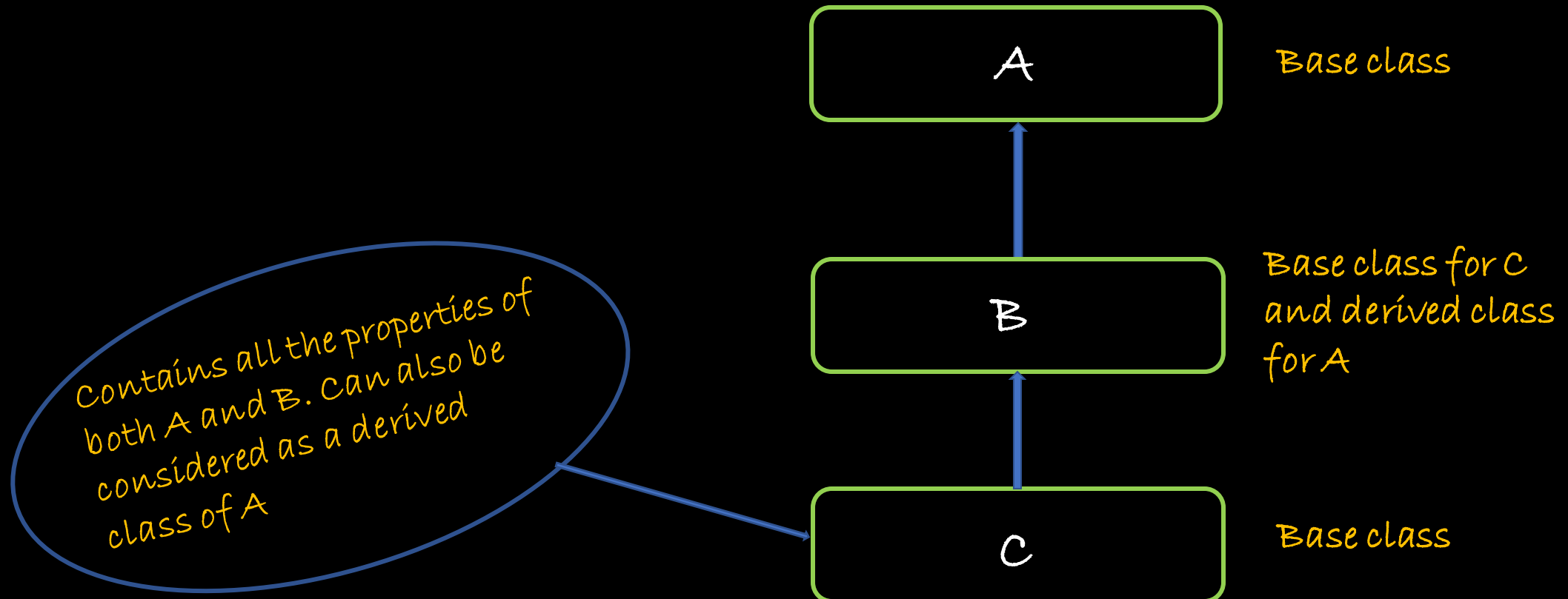


Types of Inheritance

- Single Inheritance ✓
- Multilevel Inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

Multilevel Inheritance

- It is a process where an inherited class is further made as a parent class to another class



Multilevel Inheritance Cont.

- Syntax

```
class A{  
//only class A properties;  
};
```

Use appropriate
access specifier

```
Class B : public A //child of A- intermediate base class  
{  
//contains both class A and class B properties  
};
```

```
class C : public class B //child of B  
{  
//contains properties of class A, class B and class C  
};
```

Multilevel Inheritance - Example

```
//BASE CLASS
class base {
public:
    int x;
    void getdata()
    {
        cin >> x;
    }
};
```

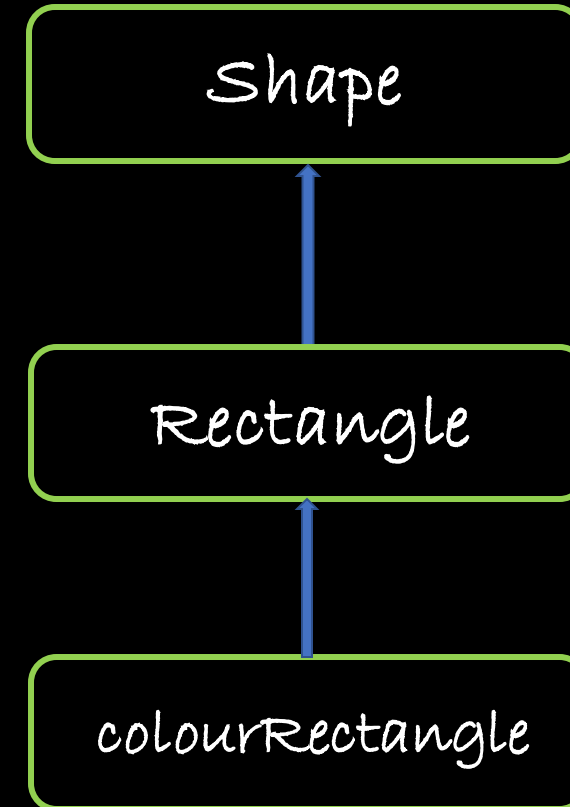
```
//DERIVED CLASS FROM BASE
class derive1: public base
{
    public:
        int y;
        void readdata()
        {
            cin >> y;
        }
};
```

```
//DERIVED CLASS FROM DERIVE1
class derive2 : public derive1
{
    private:
        int z;
    public:
        void indata()
        {
            cin >> z;
        }
        void product()
        {
            cout<<"Product= " << x*y*z;
        }
};
```

```
//Main
int main()
{
    derive2 a;
    a.getdata();
    a.readdata();
    a.indata();
    a.product();
    return 0;
}
```

Multilevel Inheritance - Exercise

- shape has length and width
- Rectangle has a method to calculate area
- ColourRectangle defines the rectangle colour properties
- All classes are encapsulated with getters and setters
- Constructors are used to initialize the data



Multilevel Inheritance – Exercise Cont.

- The main function

```
int main() {  
    colourRectangle r1("green",2,3.5);  
    cout<<"Area: "<<r1.getArea();  
    cout<<"Colour: "<<r1.getColour();  
    colourRectangle r2("blue",2,3.5);  
    cout<<"Area: "<<r2.getArea();  
    cout<<"Colour: "<<r2.getColour();  
    return 0;  
}
```


What about the effect of the Access Specifiers?

- Try it!!
- Have 3 classes A,B,C where A is the base class, B is the intermediate class and C is the sub-sub(?) class
- Try out the different combinations
 - Public, Protected, Private
 - Eg1. `class A, class B : public A, class C : public B`
 - Eg2: `class A, class B : private A, class C : public B`
 - Eg3. `class A, class B : public A, class C : private B`
 - .
 - ... And so on...

Quick Summary

- Inheritance
- Single Inheritance Recap
- Multi-Level inheritance
- Examples
- Exercises

Up Next

Inheritance in C++ - Part 3