

Inheritance



Mohammad Tasin

Agenda

- **Diamond problem.**
- **Function overriding.**
- **Function overloading.**
- **Function hiding.**
- **Base Pointer.**

Diamond problem

```
class A {  
    public:  
    A() {cout<<"Class A call"<<endl;}  
};  
  
class B1 : public A {  
    public:  
    B1() {cout<<"Class B1 call"<<endl;}  
};  
  
class B2 : public A {  
    public:  
    B2() {cout<<"Class B2 call"<<endl;}  
};  
  
class C : public B2, public B1 {  
    public:  
    C() {cout<<"Class C call"<<endl;}  
};
```

Function overriding

- **Function overriding** that means one function is same name and same signature in the different classes and all classes inherited each others. It called **function overriding**.

Example of Function overriding

```
class Parent {  
    public:  
        void Print_Name(string name)  
        { cout<<"Parent class called, "<<name<<endl; }  
};  
  
class Child : public Parent {  
    public:  
        void Print_Name(string name)  
        { cout<<"Child class called, "<<name<<endl; }  
};
```

Function overloading

- **Multiple functions sharing same name can be mapped with function call on the basis of arguments at compile time is known as Function polymorphism or Function overloading**

Function hiding

- **When a derived class defines a function with the same name as a base class function, but with different parameters, the derived class function hides the base class function.**
 - **This means that when you call the function on an object of the derived class, the derived class's function is called, not the base class's function.**
-
- ❖ **ObjectName.ClassName::FunctionName;**
 - ❖ **ObjectName.ClassName::VariableName;**

Function hiding

```
class Parent {  
    public:  
        void Print_Name(string name) {  
            cout<<"Parent class called"<<endl;  
            cout<<"Name is "<<name<<endl;  
        }  
};  
class Child : public Parent {  
    public:  
        void Print_Name() {  
            cout<<"Child class called"<<endl;  
            cout<<"Name is Hello"<<endl;  
        }  
};
```


Base Pointer

