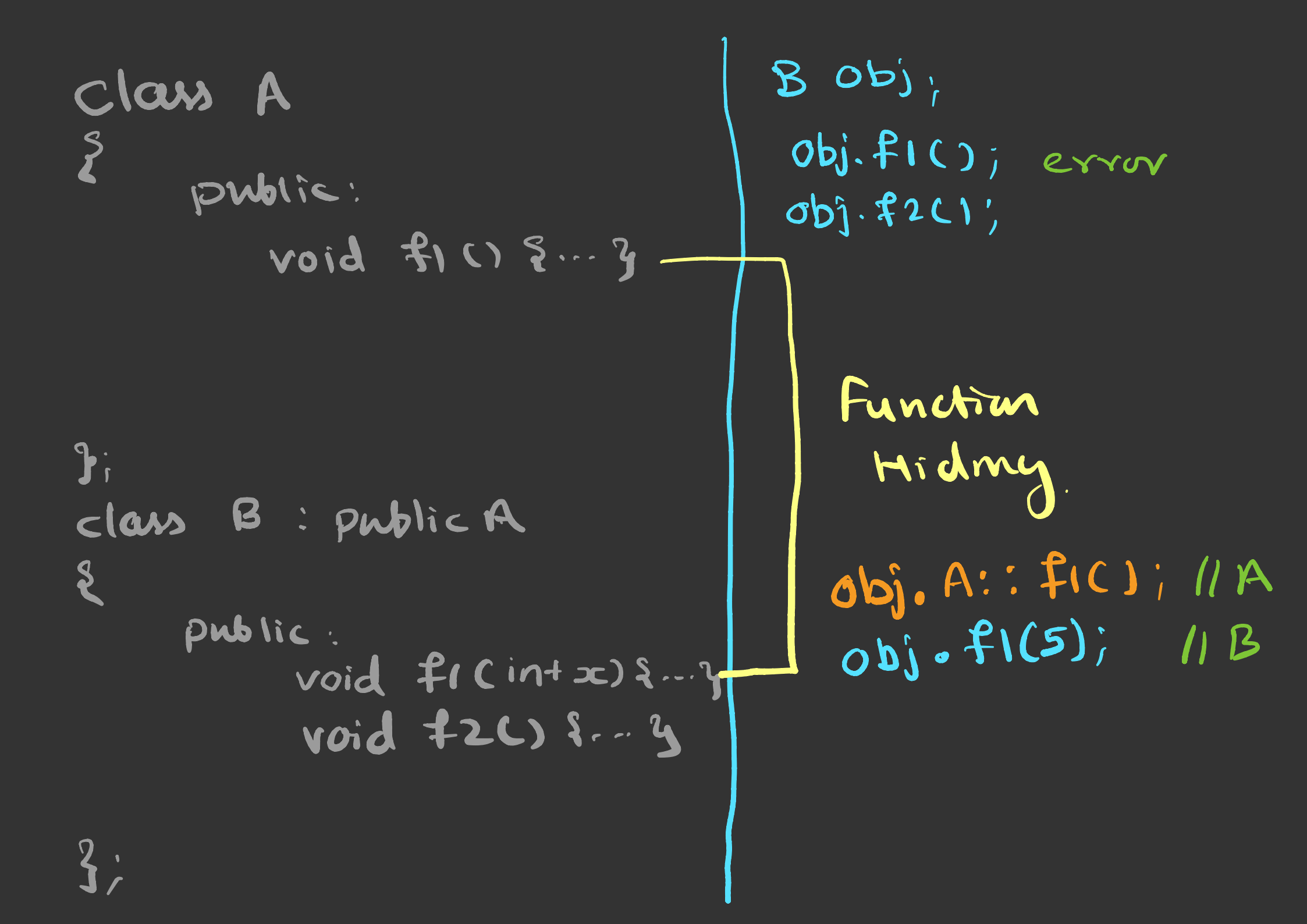
C, C++, DSA in depth

Overriding, hiding and overloading

Saurabh Shukla (MySirG)

Agenda

1. Function Overriding
2. Function Hiding
3. Function Overloading


Class A
{
public:
void f1() {... }
};
class B : public A
{
public:
void f1(int x) {... }
void f2() {... }
};
B obj;
obj.f1(); error
obj.f2();
Function Hiding.
obj.A::f1(); // A
obj.f1(5); // B

Function Overloading

Functions with the same name but differ in arguments and reside in sigle scope is known as overloaded functions, and this phenomenon is known as Function overloading or function polymorphism.

All overloaded versions of a function must be in the same class.

Function Hiding

When child class has a function with the same name but differ in arguments as compare to the version of function of parent class, we say, function in the child class hides the version of parent class. This concept is called function hiding.

One version must be in base class and the other version should be in a derived class of any of its descendant class.

```
Class A
{
    public:
        void f1() {...}

};
Class B : public A
{
    public:
        void f1() {...}

};
```

Function overriding

```
B obj;
obj.f1(); //B
```

Function Overriding

Same prototype function, one version reside in base class and other version resides in derived (or any descendant class) class, then version defined in derived class overrides the version defined in base class.

This concept is known as function overriding.

Why overriding?

```
class Car
{
    gear
  public:
  void shiftGear() {...}


};
class SportsCar : public Car
{

   void shiftGear() {...}


};
```

```
SportsCar obj;
obj.shiftGear();
```