IIT Bombay

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Access Control in Derived Classes

Recap

IIT Bombay

- Compositional vs inheritance-based approaches of representing hierarchy of classes
- Class hierarchy
  - Base/super class
  - Derived class
  - All members were public
  - Inheritance/derivation was public (class D: public class B)

Overview of This Lecture

IIT Bombay

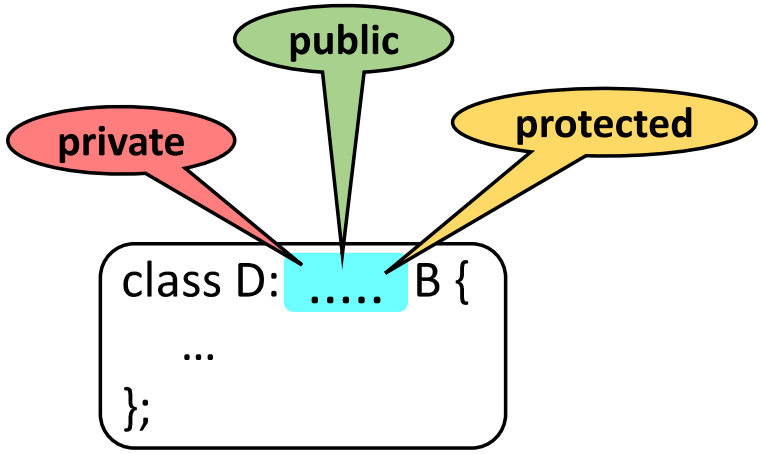
- Inheritance with public, private and protected members
- Public, private and protected inheritance/derivation
- Access control in derived classes

Class Inheritance Mechanism

IIT Bombay

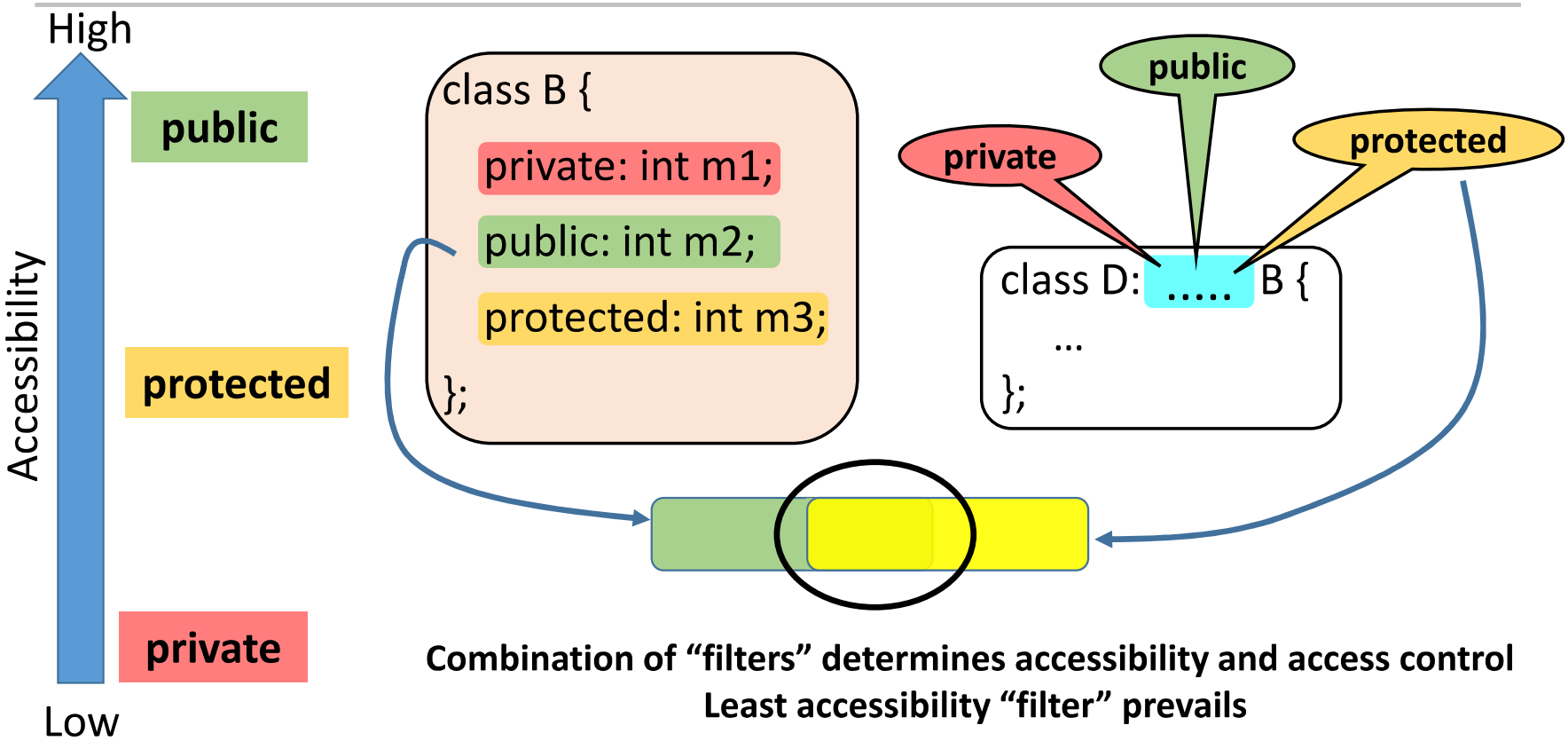
```
class B {
    private: int m1;
    public: int m2;
    protected: int m3;
};
```

"Filters" to control access to members



"Filters" to control inheritance

Class Inheritance Mechanism



Combination of "filters" determines accessibility and access control
Least accessibility "filter" prevails

Derived Class Type

IIT Bombay

```
class B {
    private: int m1;
    public: int m2;
    protected: int m3;
};
```

```
class D : public B{
  int temp;
};
```

```
int main(){
    D d;
    cin >> d.m1;
    cin >> d.m2;
    cin >> d.m3;

    B b;
    cout << b.m1;
    cout << b.m2;
    cout << b.m3;
    return 0;
}
```

- ✗ cannot access private member data of the base class
- ✓ can access public member data of the base class
- ✗ cannot access protected member data of the base class
- ✗ cannot access private data member
- ✓ can access public data member
- ✗ cannot access protected data member

Derived Class Type (Class D1)

IIT Bombay

```
int main() {
  D1 d1;
  d1.f1 ();
  d1.f2 ();
  d1.m1 = 1;
  d1.m2 = 1;
  return 0;
};
```

can access all the data members through the member function 'f1()' of the same class

can access only public and protected members of the base class

```
class D1: public  B {
  private:   int x1;
  public:    int x2;
  protected: int x3;
  public:
    void f1(){
      cin >> x1;
      cin >> x2;
      cin >> x3;
    }
    void f2(){
      cin >> m1;
      cin >> m2;
      cin >> m3;
    }
};
```

public: int m2;

protected: int m3;

Can access all members of its class

```
class B {
  private: int m1;
  public: int m2;
  protected: int m3;
public:
  void k(){
    cin >> m1;
    cin >> m2;
    cin >> m3;
  }
};
```

Base Class Members	Public Derivation
Private	Not directly accessible
Protected	Protected
Public	Public

Table (a)

m1 cannot be directly accessed

m2 inherited as public member

m3 inherited as a protected member

Derived Class Type (Class D2)

IIT Bombay

```
int main() {
  D2 d2;
  d2.h1 ();
  d2.h2 ();
  d2.m2 = 1;
  return 0;
};
```

can access all the data members through the member function 'h1()' of the same class

can access only public and protected members of the base class, both these members become protected

Base Class Members	Protected Derivation
Private	Not directly accessible
Protected	Protected
Public	Protected

Table (b)

Can access all members of its class

m1 cannot be directly accessed

m2 inherited as protected member, originally public in base

m3 inherited as protected member

```
class D2:  protected  B {
  private:   int z1;
  public:    int z2;
  protected: int z3;
  public:
    void h1(){
      cin >> z1;
      cin >> z2;
      cin >> z3;
    }
    void h2(){
      cin >> m1;
      cin >> m2;
      cin >> m3;
    }
};
```

protected: int m2;

protected: int m3;

```
ss B {
rivate: int m1;
ublic: int m2;
rotected: int m3;
blic:
void k(){
  cin >> m1;
  cin >> m2;
  cin >> m3;
}
```

Derived Class Type (Class D3)

IIT Bombay

```
int main() {
  D3 d3;
  d3.g1 ();
  d3.g2 ();
  d3.m2 = 1;
  return 0;
};
```

can access all the data members through the member function 'g1()' of the same class

can access only public and protected members of the base class, both these members become private

```
class D3: private B {
  private:   int y1;
  public:    int y2;
  protected: int y3;
  public:
    void g1(){
      cin >> y1;
      cin >> y2;
      cin >> y3;
    }
    void g2(){
      cin >> m1;
      cin >> m2;
      cin >> m3;
    }
};
```

private: int m2;

private: int m3;

```
class B {
  private: int m1;
  public: int m2;
  protected: int m3;
public:
  void k(){
    cin >> m1;
    cin >> m2;
    cin >> m3;
  }
};
```

Base Class Members	Private Derivation
Private	Not directly accessible
Protected	Private
Public	Private

Table (c)

Can access all members of its class

m1 cannot be directly accessed

m2 inherited as private member, originally public in base

m3 inherited as a private member, originally protected in base

Visibility of Base Class Members (Table)

IIT Bombay

Base Class Members	Public Derivation	Protected Derivation	Private Derivation
Private	Not directly accessible	Not directly accessible	Not directly accessible
Protected	Protected	Protected	Private
Public	Public	Protected	Private

Table (a) from Class D1

Table (b) from Class D2

Table (c) from Class D3

Inheritance of Member Functions

IIT Bombay

```
int main() {
  B b;
  b.g2();

  D1 d1;
  d1.g1();
  d1.f1();
  d1.f2();

  D2 d2;
  d2.g1();
  d2.h1();
  d2.h2();

  return 0;
};
```

cannot be accessed from 'main()', as member function 'b.g2()', 'd2.g1()', 'd1.f2()', 'd2.h2()' are protected

```
class D2: protected D1 {
  private:   int y1;
  public:    int y2;
  protected: int y3;
  public:
    void h1() {
      g1();
      g2();
      f1();
      f2();
    }
  protected:
    void h2() {
      g1();
      g2();
      f1();
      f2();
    }
};
```

protected: int m2
protected: int m3

protected: int x2
protected: int x3

protected: g1()
protected: g2()
protected: f1()
protected: f2()

```
class D1: public B {
  private:   int x1;
  public:    int x2;
  protected: int x3;
  public:
    void f1() {
      g1();
      g2();
    }
  protected:
    void f2(){
      cin >> x1;
      cin >> x2;
      cin >> x3;
    }
};
```

public: int m2;

protected: int m3;

public: void g1()

protected: void g2()

```
class B {
  private: int m1;
  public: int m2;
  protected: int m3;
 public:
    void g1() {
      cout << "h";
    }
    protected:
    void g2(){
      cin >> m1;
      cin >> m2;
      cin >> m3;
    }
};
```

Summary

IIT Bombay

- Access control in derived classes
- Inheritance with public, protected and private members
- Public, protected, and private inheritance