

Comp151

Public, Protected & Private Inheritance

Back to Inheritance

- Before we saw public inheritance of the form
class Student : public person {
- We will now see *protected* and *private* inheritance as well.
- They differ in how the inherited members of Student are accessed by classes that are derived from Student.
- Note: Public inheritance is the norm. Private and protected inheritance are very unusual.

Example: person.h

```
class Person {
private:
    string name;
    string address;
    Department dept;
protected:
    void set_name(const char *name);
    void set_address(const char *adr);
    void set_department(Department dept);
public:
    Person(string n, string a, Department d) :
        name(n), address(a), dept(d) { };
    string get_name() const;
    string get_address() const;
    Department get_department() const;
};
```

Example: student.h

```
class Student : ??? Person
{ private:
    Course *enrolled;
    int num_courses;
public:
    Student(string n, string a, Department d) :
        Person(n, a, d), enrolled(NULL), num_courses(0) { }
    bool enroll_course(const string &);
    bool drop_course(const Course &);
};
```

- If GraduateStudent is derived from Student what members of Student can GraduateStudent access?

Example: Public Inheritance

```
class Student: public Person { ... }
```

public	protected	private	not accessible
get_name()	set_name()	enrolled	name
get address()	set_address()	num_courses	address
get_department()	set_department()		dept
enroll_course()			
drop_course()			

Example: Protected Inheritance

```
class Student: protected Person { ... }
```

public	protected	private	not accessible
enroll_course()	set_name()	enrolled	name
drop_course()	set_address()	num_courses	address
	set_department()		dept
	get_name()		
	get_address()		
	get_department()		

Example: Private Inheritance

class Student: **private** Person { ... }

public	protected	private	not accessible
enroll_course()		enrolled	name
drop_course()		num_courses	address
		set_name()	dept
		set_address()	
		set_department()	
		get_name()	
		get_address()	
		get_department()	

Summary

- Public inheritance preserves the original accessibility of the base class' public and protected members:
 - public => public
 - protected => protected
 - private => not accessible
- Protected inheritance causes the accessibility of inherited public and protected members to be protected.
 - public => protected
 - protected => protected
 - private => not accessible

Summary...

- Private inheritance renders all inherited members private.
 - public => private
 - protected => private
 - private => not accessible
- Remember that the base class' private members are never accessible to derived classes (or to any other classes).
- The various types of inheritance control the highest accessibility of the inherited member data and functions.
- Public inheritance is the most common form of inheritance; protected and private inheritance are rarely used these days.