Comp151

Public, Protected & Private Inheritance

Back to Inheritance

Before we saw *public inheritance* of the form

$$\textbf{class}\ \text{Student} : \textbf{public}\ \text{person}\ \{\}$$

We will now see *protected* and *private* inheriteace 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
get_name()	set_name()	name
get_address()	set_address()	address
get_department()	set_department()	dept
enroll_course()		enrolled
drop_course()		num_courses

Example: Protected Inheritance

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

public	protected	private
enroll_course()	set_name()	name
drop_course()	set_address()	address
	set_department()	dept
	get_name()	enrolled
	get_address()	num_courses
	get_department()	

Example: Private Inheritance

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

public	protected	private
enroll_course()		name
drop_course()		address
		dept
		enrolled
		num_courses
		set_name()
		set_address()
		set_department()
		get_name()
		get_name()
		get_address()
		get_department()

Summary

- Public inheritance *preserves* the original accessibility of inherited members:

$$\begin{aligned} \text{public} &\Rightarrow \text{public} \\ \text{protected} &\Rightarrow \text{protected} \\ \text{private} &\Rightarrow \text{private} \end{aligned}$$

- Protected inheritance affects only public members and renders them protected.

$$\begin{aligned} \text{public} &\Rightarrow \text{protected} \\ \text{protected} &\Rightarrow \text{protected} \\ \text{private} &\Rightarrow \text{private} \end{aligned}$$

Summary ..

- Private inheritance renders all inherited members private.

$$\begin{array}{rcl} \text{public} & \Rightarrow & \text{private} \\ \text{protected} & \Rightarrow & \text{private} \\ \text{private} & \Rightarrow & \text{private} \end{array}$$

- The various types of inheritance control the highest accessibility of the <u>inherited member</u> data and functions.
- Public inheritance implements the "is-a" relationship.
- Private inheritance is similar to "has-a" relationship.
- Public inheritance is the most common form of inheritance.