10. Inheritance: *Encapsulation & Access*

private, public & protected members

```cpp
class Manager : public Employee {
    Employee* managed_list;
protected:
    int level;
public:
    int getLevel();
};

class Director : public Manager {
    ...
    void fn(......)
    {
        level = UPPER_MANAGEMENT; // OK
        groupsize =
            managed_list.size(); // ERROR
        getLevel(); // OK
    }
};

Director laura;
laura.getLevel(); // OK
laura.level = UPPER_MANAGEMENT; // ERROR
```
Access Control in Inheritance

Base Part of Object

Derived Part of Object

From Outside the Object

- Private member
- Protected member
- Public member

Inheritance: private, public & protected

class Manager : public Employee {...};

class WindowWDlg : private Dialog {...};

public Inheritance:
  • Truly expresses a “is-a” relationship
protected Inheritance :
  • Expresses inheritance for data reuse and not behavior reuse
  • This is better modeled as Containment.
Inheritance Control - Example

class Employee {...};
class Manager : public Employee {...};
class Supervisor : private Employee {...};
void SecurityCheck(const Employee&);

Employee bruce;
Manager nancy;
Supervisor susan;

SecurityCheck(bruce); // OK
SecurityCheck(nancy); // OK
SecurityCheck(susan); // ERROR

Lab Work: Details provided on-line.