Inheritance

1. Inheritance

Inheritance is when a class is derived from another class so that it “inherits” attributes from its “parent” class.

For example, rectangles and triangles are both polygons. They both have common features such as the height and width, but they have different features such as the calculation of their area. So we can create two classes Rectangle and Triangle that inherit from the parent class Polygon.

```
// derived classes
#include <iostream>
using namespace std;

////////////////// Parent class - Polygon /////////////////////////
class Polygon {
   protected:
      int width, height;
   public:
      void set_values (int a, int b) {
         width=a; height=b;
      }
};

////////////////// Derived class - Rectangle /////////////////////////
class Rectangle: public Polygon {
   public:
      int area () {
         return (width * height);
      }
};

////////////////// Derived class - Triangle /////////////////////////
class Triangle: public Polygon {
   public:
      int area () {
         return (width * height / 2);
      }
};
```
```cpp
int main () {
    Rectangle rect;
    Triangle trgl;
    rect.set_values (4,5);
    trgl.set_values (4,5);
    cout << rect.area() << endl;
    cout << trgl.area() << endl;
    return 0;
}
```

The objects of the classes Rectangle and Triangle each contain members inherited from Polygon. These are: width, height and set_values().

The protected access specifier is similar to private. Its only difference occurs in fact with inheritance. When a class inherits from another one, the members of the derived class can access the protected members inherited from the base class, but not its private members. Non-class members cannot access the protected members.
2. Exercises (Problems with an asterisk are more difficult)

1. Write some code to confirm that non-class members cannot access the protected members.

2. Change “protected” to “private” in the parent class and confirm that a derived class cannot access the private members.

3. Write a program to create the following parent and derived classes. The members of each class are specified.

   - **Person**
     - Protected:
       - Name
       - Address
       - Phone

   - **Employee**
     - Protected:
       - Department
       - Date of hire

   - **Student**
     - Protected:
       - Major
       - Year
       - GPA

   - **Faculty**
     - Private:
       - Rank
       - Salary

   - **Staff**
     - Private:
       - Hourly Rate

Include code to set values and get values for each member field. Test it out with a main program.