EEE102 C++ Programming and Software Engineering II

Lab Practice 8

Friendship and Inheritance

Notice:

- The aim of this lab is for you to become familiar with the usage of the keyword `friend` and class inheritance.
- Practice with the exercises. These parts are not for submission.

1. friend

Exercise 1.1

Answer the questions below.

1

Is this declaration of friendship correct or not?

```
class classA
{
private:
    friend class classB;
public:
    . . . . . .
};
```

2

Read the piece of program given below:

```
class classA
{
    friend class classB;
    . . . . . .
};
```

```
class classB
{
    friend class classC;
    . . . . . .
};
```

Are the following statements correct?

a) classA is classB's friend, so classA can use classB's private member;
b) classC is classB's friend, so classC can use classB's private member;
c) classB is classA's friend, classC is classB's friend, so classC is classA's friend;
d) classB is classA's friend, so subclasses derived from classB are also classA's friends.

3

What are the specifiers?

Explain the difference between **`public`**, **`protected`** and **`private`**.

2. Class inheritance

Exercise 2.1

Read the piece of program given below, find out the accessibility of the class members, and fill in the table shown below.

```
class clbase
{
public:
    clbase();
    ~clbase();
    int b;
    char c();
protected:
    int d;
private:
    char f;
};
```

```
class clsub : protected clbase
{
public:
    double g();
private
    int h;
};
```

Fill in the table shown below:

Member names	Access from outside	Inherited by subclass	Access inside itself
`clbase::b`	Yes		
`clsub::c()`			
`clbase::d`			
`clsub::d`	No		
`clbase::f`			
`clsub::f`			
`clsub::g()`			
`clsub::h`			

Exercise 2.2

Read the piece of program given below, find out the accessibility of the class members, and answer the following questions.

```
class Point
{
public:
    Point (int a=0, int b=0);
    ~ Point ();
    void setXY(int a, int b);
    void setFlag(bool f);
    bool getFlag();
    void display();
protected:
    int x,y;
private:
    bool flag;
};
```

```
class ThreeDPoint : public Point
{
public:
    void setZ(int c);
    void display();
private:
    int z;
};
```

```
class FourDPoint : public ThreeDPoint
{
public:
    void setW(int d);
    void display();
private:
    int w;
};
```

```
class FiveDPoint : protected FourDPoint
{
public:
    void setV(int e);
    void display();
private:
    int v;
};
```

Are the following statements (in the main function) correct or not? Why?

1	`Point myp2;`
2	`myp2.x=0; myp2.y=0;`
3	`myp2.setFlag(1);`
4	`ThreeDPoint myp3(3,4);`
5	`ThreeDPoint.setZ(5);`
6	`myp3.getFlag;`
7	`FourDPoint myp4;` `myp4.setXY(1,2);`
8	`myp4.setZ(5);`
9	`myp4.setW(6);`
10	`FiveDPoint myp5;` `myp5.setXY(10,20);`

Exercise 2.3 In this part you are asked to consider the concept of transport. If you are going to design a programme which models the different types of transport available, how would you do it?	
1.	Start with a class called transport. Produce a class diagram showing two typical methods (function members) and two associated properties (data members) that are common to all modes of transport. Specify the access specifier for all data members and function members so that the function members **cannot** be accessed from outside the class but can be inherited by any sub-class. The properties identified here must be applicable to all transport types.
2	Next derive a set of sub-classes which represent different means of transport (air, land, water --second level). For each of these sub-classes, also produce a class diagram showing two typical methods (function members) and two associated properties (data members) specific to the sub-class. Specify the access specifier for all data members and function members so that the function members **cannot** be accessed from outside the class but can be inherited by any sub-class. The properties identified here must be specific to each means of transport.
3	Next generate further lower level sub-classes (more than 1 level) including: car, truck, bus, train, horse, fishing boat, submarine, spaceship and jet passenger plane. For each of these sub-classes, also produce a class diagram showing two typical methods (function members) and two associated properties (data members) specific to the sub-class. Specify the access specifier for all data members and function members so that the function members can be accessed from outside the class this time. The properties identified here must be specific to each type of transport.
4	Finally organise all the classes into a hierarchy chart and specify the inheritance specifier for each sub-class.