

1. There is a student class, including the student's name, grades, and requirements:

- (1) Design a friend function compare() to compare the grade of two students.
- (2) Define an object array in the main() function to store the input student data, and find the student with the highest score and the lowest score.

2. Define an abstract class Shape, including a pure virtual function perimeter, whose function is to calculate the perimeter of the graph. There are three classes derived from Shape:

- (1) Triangle, the data member includes the length of three sides (a, b, c);
- (2) Rectangle (rectangle), data members include width (width) and height (height);
- (3) Circle, the data member is a radius.

Reasonably use polymorphism to realize the perimeter function to calculate the perimeter of triangles, rectangles, and circles.

Create triangle objects, rectangle objects and circle objects in main () and display their information.

3. Define the Person class. The Teacher class and the Cadre class are derived from Person. Requirements are as follows:

(1) Data members:

- 1. The data members included in the Person class include name, age, and gender.
- 2. Include data members in the Teacher class: position.
- 3. Include data members in the Cadre class: work, salary.

(2) Member function:

- 1. Define member functions in the class body (including construction and destructor functions according to the actual situation)
- 2. Each class has a display information function (Show)
- 3. Create a Teacher object and a Cadre object in main() and display their information.

4. Define a base class Animal, which contains two private data members, one is the name of the animal (for example, "Fido" or "Yogi"), and the other is the weight of the animal.

This class also contains a public member function who(), which can display the animal's name and weight. Using Animal as a public base class, derive two classes Lion and Aardvark.

Write a main() function to create Lion and Aardvark objects ("Leo", 400 points; "Algernon", 50 points). Calling the who() member for the derived class object indicates that the who() member is inherited in the two derived classes.