1. Create two classes named Mammals and MarineAnimals. Create another class named BlueWhale which inherits both the above classes. Now, create a function in each of these classes which prints "I am mammal", "I am a marine animal" and "I belong to both the categories: Mammals as well as Marine Animals" respectively. Now, create an object for each of the above class and try calling

    1 - function of Mammals by the object of Mammal

    2 - function of MarineAnimal by the object of MarineAnimal

    3 - function of BlueWhale by the object of BlueWhale

    4 - function of each of its parent by the object of BlueWhale

2. Make a class named Fruit with a data member to calculate the number of fruits in a basket. Create two other class named Apples and Mangoes to calculate the number of apples and mangoes in the basket. Print the number of fruits of each type and the total number of fruits in the basket.

3. We want to calculate the total marks of each student of a class in Physics,Chemistry and Mathematics and the average marks of the class. The number of students in the class are entered by the user. Create a class named Marks with data members for roll number, name and marks. Create three other classes inheriting the Marks class, namely Physics, Chemistry and Mathematics, which are used to define marks in individual subject of each student. Roll number of each student will be generated automatically.

4. We want to store the information of different vehicles. Create a class named Vehicle with two data member named mileage and price. Create its two subclasses

    *Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol).

    *Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches)

    Make another two subclasses Audi and Ford of Car, each having a data member to store the model type. Next, make two subclasses Bajaj and TVS, each having a data member to store the make-type.

Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.) Do the same for a Bajaj and a TVS bike.

5. Create a class named Shape with a function that prints "This is a shape". Create another class named Polygon inheriting the Shape class with the same function that prints "Polygon is a shape". Create two other classes named Rectangle and Triangle having the same function which prints "Rectangle is a polygon" and "Triangle is a polygon" respectively. Again, make another class named Square having the same function which prints "Square is a rectangle".

   Now, try calling the function by the object of each of these classes.

6. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in Fig. 8.14. The figure also shows the minimum information required for each class. Specify all the classes and define functions to create the database and retrieve individual information as and when required

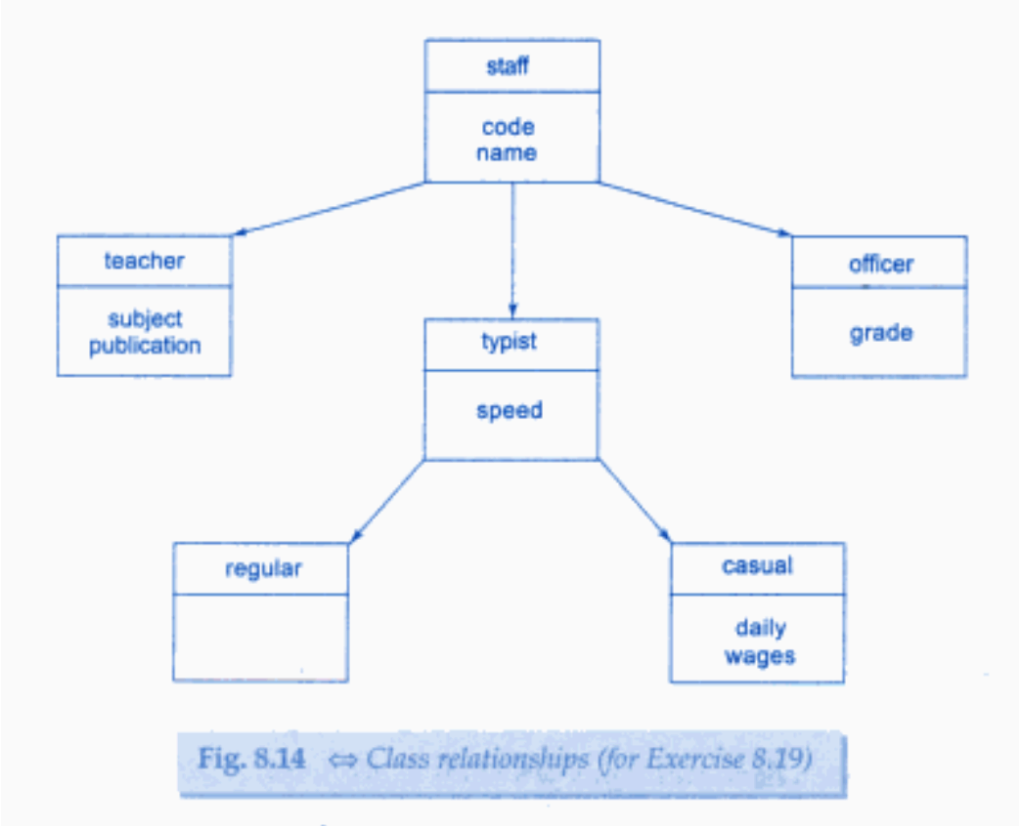

Fig. 8.14 ⇔ *Class relationships (for Exercise 8.19)*

7. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data()

to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.
Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.
Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles, and used as follows:

Area of rectangle= x* y
Area of triangle= ½*x*y

8. Extend the above program to display the area of circles. This requires addition of a new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only one value, its radius, but the get_data() function in the base class requires two values to be passed. (Hint: Make the second argument of get_data() function as a default one with zero value)

9. Write a program with the following:
   (a) A function to read two double type numbers from keyboard
   (b) A function to calculate the division of these two numbers
   (c) A try block to throw an exception when a wrong type of data is keyed in
   (d) A try block to detect and throw an exception if the condition "divide-by-zero occurs
   (e) Appropriate catch block to handle the exceptions thrown

10. Student is a class with two private member variables name and marks, and a public void member function showdata(). Create two objects named ‘s1’and ‘s2’ in the main function. Overload the operators '++' and ‘--’ in this class to perform the increment of marks and decrement of marks through the following instructions ++s1 and s2-- from the main function.

11. Create an employee hierarchy with a base class Employee and derived classes such as Manager, Engineer, Salesperson, etc. Each employee should have a calculateSalary() method. Implement this method differently for each employee type and demonstrate polymorphism by calling calculateSalary() on objects of different employee types.

12. Create an abstract shape drawing application where users can draw various shapes on a canvas. Define an abstract base class **Drawable** containing pure virtual functions for drawing shapes and moving them on the canvas. Implement derived classes such as **Circle**, **Rectangle**, and **Triangle** where each class overrides the pure virtual functions to draw the respective shapes on the canvas.