









Polymorphism, Interface, abstract: Payroll System Modification

﻿

8.4 (Rectangle Class) Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide methods that calculate the rectangle's perimeter and area. It has set and get methods for both length and width. The set methods should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0. Write a program to test class Rectangle. 8.5 (Modifying the Internal Data Representation of a Class) It would be perfectly reasonable for the Time2 class of Fig. 8.5 to represent the time internally as the number of seconds since mid- night rather than the three integer values hour, minute and second. Clients could use the same pub- lic methods and get the same results. Modify the Time2 class of Fig. 8.5 to implement the time as the number of seconds since midnight and show that no change is visible to the clients of the class.

8.6 (Savings Account Class) Create class Savings Account. Use a static variable annual Inter- estrate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthly Interest to calculate the monthly interest by multiplying the savingsBalance by annual InterestRate divided by 12—this interest should be added to savings- Balance. Provide a static method modifyInterestRate that sets the annual InterestRate to a new value. Write a program to test class Savings Account. Instantiate two savings Account objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set annual InterestRate to 4%, then calculate the monthly interest for each of 12 months and print the new balances for both savers. Next, set the annual InterestRate to 5%, calculate the next month's interest and print the new balances for both savers.

8.7

(Enhancing Class Time2) Modify class Time2 of Fig. 8.5 to include a tick method that in- crements the time stored in a Time2 object by one second. Provide method incrementMinute to in- crement the minute by one and method incrementHour to increment the hour by one. Write a program that tests the tick method, the incrementMinute method and the incrementHour method to ensure that they work correctly. Be sure to test the following cases:

a) incrementing into the next minute,

b) incrementing into the next hour and

c) incrementing into the next day (i.e., 11:59:59 PM to 12:00:00 AM)..

﻿

9.8 (Quadrilateral Inheritance Hierarchy) Write an inheritance hierarchy for classes Quadri- lateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Create and use a Point class to represent the points in each shape. Make the hier- archy as deep (i.e., as many levels) as possible. Specify the instance variables and methods for each class. The private instance variables of Quadrilateral should be the x-y coordinate pairs for the four endpoints of the Quadrilateral. Write a program that instantiates objects of your classes and outputs each object's area (except Quadrilateral).

﻿

9.10

(Write a Line of Code) Write a line of code that performs each of the following tasks:

a) Specify that class PieceWorker inherits from class Employee.

b) Call superclass Employee's toString method from subclass Pieceworker's toString method.

c) Call superclass Employee's constructor from subclass PieceWorker's constructor-as- sume that the superclass constructor receives three Strings representing the first name, last name and social security number.

﻿

7.10

(Sales Commissions) Use a one-dimensional array to solve the following problem: A com- pany pays its salespeople on a commission basis. The salespeople receive $200 per week plus 9% of their gross sales for that week. For example, a salesperson who grosses $5,000 in sales in a week re- ceives $200 plus 9% of $5,000, or a total of $650. Write an application (using an array of counters) that determines how many of the salespeople earned salaries in each of the following ranges (assume that each salesperson's salary is truncated to an integer amount):

a) $200-299

b) $300-399 c) $400-499 d) $500-599 e) $600-699 f) $700-799 g) $800-899 h) $900-999

i) $1,000 and over

Summarize the results in tabular format.

7.11 Write statements that perform the following one-dimensional-array operations:

a) Set the 10 elements of integer array counts to zero.

b) Add one to each of the 15 elements of integer array bonus.

c) Display the five values of integer array bestScores in column format.

7.12 (Duplicate Elimination) Use a one-dimensional array to solve the following problem: Write an application that inputs five numbers, each between 10 and 100, inclusive. As each number is read, display it only if it's not a duplicate of a number already read. Provide for the "worst case," in which all five numbers are different. Use the smallest possible array to solve this problem. Display the complete set of unique values input after the user enters each new value.

7.13 Label the elements of three-by-five two-dimensional array sales to indicate the order in which they're set to zero by the following program segment:

for (int row = 0; row < sales.length; row++)

{

for (int col = 0; col< sales [row].length; col++)

{

sales[row][col] = 0;

}

}

7.14 (Variable-Length Argument List) Write an application that calculates the product of a series of integers that are passed to method product using a variable-length argument list. Test your meth- od with several calls, each with a different number of arguments.