## **Object-Oriented Programming**

The assignments are Programming Exercises 6.23 on pages 241-242, 7.9 on page 280, 8.1 on pages 307-308, 9.1 on page 362, and 11.1 on page 447 of the textbook (Chapters 6-9, 11). For your convenience, the exercises are shown below, where 9.1 and 11.1 are modified slightly. Please copy here your source codes, including your input and output screenshots. Please upload this document along with your source files (i.e., the .java files) on Blackboard by the due date.

6.23 (*Occurrences of a specified character*) Write a method that finds the number of occurrences of a specified character in a string using the following header:

public static int count (String str, char a)

For example, count ("Welcome", 'e') returns 2. Write a test program that prompts the user to enter a string followed by a character then displays the number of occurrences of the character in the string.





7.9 (*Find the smallest element*) Write a method that finds the smallest element in an array of double values using the following header:

public static double min (double [] array)

Write a test program that prompts the user to enter 10 numbers, invoke this method to return the minimum value, and displays the minimum value. Here is a sample run of the program:

Enter 10 numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2

The minimum number is 1.5



8.1 (*Sum elements column by column*) Write a method that returns the sum of all the elements in a specified column in a matrix using the following header:

public static double sumColumn (double [][] m, int columnIndex)

Write a test program that reads a 3-by-4 matrix and displays the sum of each column. Here is a sample run:

Enter a 3-by-4 matrix row by row:

1.5234

5.5678

9.5131

Sum of the elements at column 0 is 16.5

Sum of the elements at column 1 is 9.0

Sum of the elements at column 2 is 13.0

Sum of the elements at column 3 is 13.0



9.1 (*The Rectangle class*) Following the example of the Circle class in Section 9.2, design a class named Rectangle to represent a rectangle. The class contains:

- Two double data fields named width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.
- A no-arg constructor that creates a default rectangle.
- A constructor that creates a rectangle with the specified width and height.
- A method named getArea () that returns the area of this rectangle.
- A method named getPerimeter () that returns the perimeter.

Write a test program that creates two Rectangle objects — one with width 4 and height 40, and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this order.

11.1 (*The Triangle class*) Design a class named Triangle that extends GeometricObject defined in Listing 11.1. The class contains:

- Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of a triangle.
- A no-arg constructor that creates a default triangle.

- A constructor that creates a triangle with the specified side1, side2, and side3.
- The accessor methods for all three data fields.
- A method named getArea () that returns the area of this triangle.
- A method named getPerimeter () that returns the perimeter of this triangle.
- A method named toString () that returns a string description for the triangle.

The formula for computing the area of a triangle is

$$s = (side1 + side2 + side3)/2;$$
  
$$area = \sqrt{s(s - side1)(s - side2)(s - side3)}$$

The toString () method is implemented as follows:

return "Triangle: side1 = " + side1 + " side2 = " + side2 + " side3 = " + side3;

Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a Triangle object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not.