Programming 2

****

Notice: Your ID and name should be printed before the outputs of all programs.

****

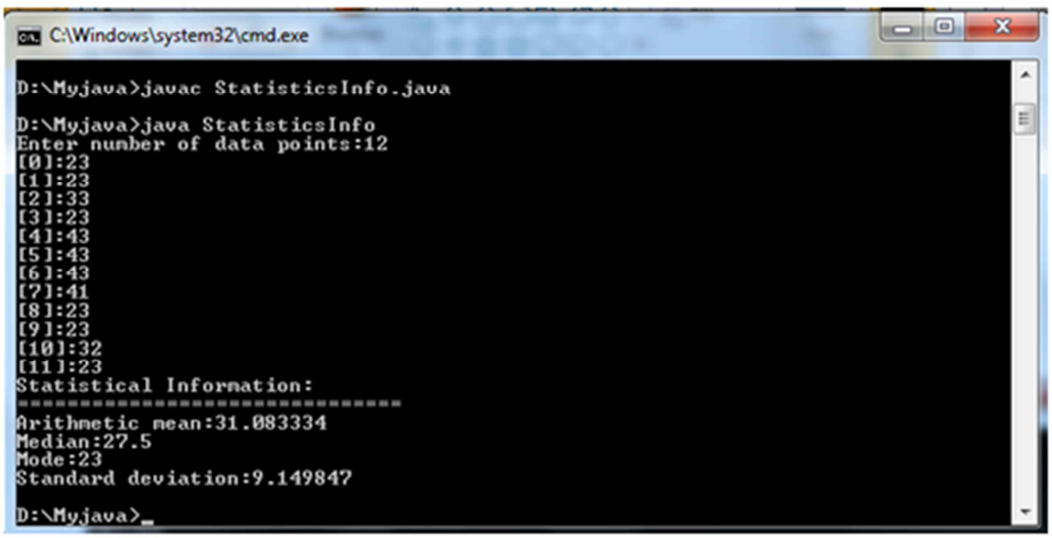
1. Write a program that multiplies the arguments and return the result as follows. You should use the concept of overloading in your program.

   mul(2, 3) ➔ 6

   mul(2, 3, 4) ➔ 24

   mul(1.5, 2.5) ➔ 3.75

2. Write a Java program to answer about the statistical information such as arithmetic mean, median, mode, and standard deviation of an integer data set. (The arithmetic median, median, and mode mean that the arithmetic average, mid-point of the distribution, and most frequent response, respectively.) The input data can be generated by random method or keyboard. Your program should display the output similar to the one shown below.



3. Write a function that simulates rolling a pair of dice until the total on the dice comes up to be a given number. The number that you are rolling for is a parameter to the function. The number of times you have to roll the dice is the return value of the function. The parameter should be one of the possible totals: 2, 3, ..., 12. Use your function in a program that computes and prints the number of rolls it takes to get snake eyes. (Snake eyes means that the total showing on the dice is 2.)

4. Write a program that will read a sequence of positive numbers entered by the user and will print the same numbers in sorted order from smallest to largest. The user will input a zero to mark the end of the input. Assume that at most 100 positive numbers will be entered. You should explain your sorting algorithm in your report.

5. Assume that you make a recipe for cooking. Design a class hierarchy that explains the foods such as pizza, noodle, pasta, meat, and coffee. Your design should include abstract classes. Write a program that prints class name, instance variables, and methods of all classes. You should make at least 10 food classes. (Note: Draw your class hierarchy in your document.)

6. Write a program that uses the methods of ArrayList. First, add 10 numbers (between 1 and 100) generated by random to list. Your program should provide three features: a(add), r(remove), and s(search). The user will input a zero to mark the end of the input. The following is an example.

   List: 16 31 25 41 1 5 19 91 35 8

   Input your command : a

   Input number to add: 67

   List: 16 31 25 41 1 5 19 91 35 8 67

Input your command : r

Input number to add: 1

List: 16 31 25 41 5 19 91 35 8 67

Input your command : s

Input number to search: 25

Index of 25 is 2.

7. Implement the Shape hierarchy shown in Fig. 1. Each TwoDimensionalShape should contain method getArea to calculate the area of the two-dimensional shape. Each ThreeDimensionalShape should have methods getArea and getVolume to calculate the surface area and volume, respectively, of the three-dimensional shape. Create a program that uses an array of Shape references to objects of each concrete class in the hierarchy. The program should print a text description of the object to which each array element refers. Also, in the loop that processes all the shapes in the array, determine whether each shape is a TwoDimensionalShape or a ThreeDimensionalShape. If it's a TwoDimensionalShape, display its area. If it's a ThreeDimensionalShape, display its area and volume.

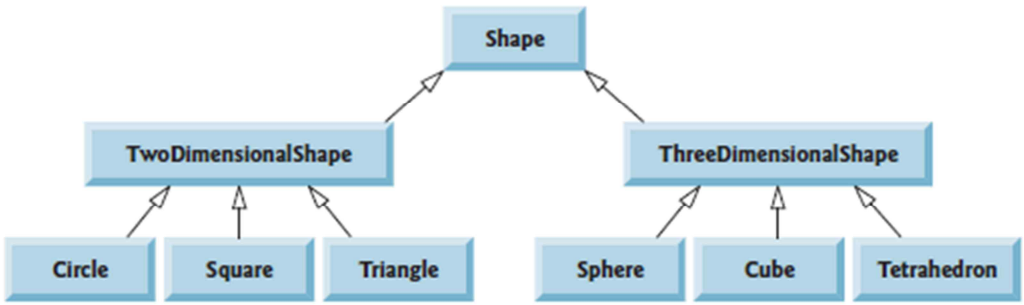


Fig. 1 Inheritance hierarchy for Shapes.