Lab 15 (October 30 or October 31)

Instructions: Complete the exercises below. Be sure to show your code to one of the lab TAs before you leave, so that you can receive credit for this lab. You must also upload a copy of all your source code (.java) files to the link on Blackboard by 11:59 PM on Tuesday, October 31.

1. A complex number is a number in the form a + bi, where a and b are real numbers and i is sqrt(-1). The numbers a and b are known as the real part and imaginary part of the complex number, respectively. You can perform addition, subtraction, multiplication, and division for complex numbers using the following formulas:

    a + bi + c + di = (a + c) + (b + d)i

    a + bi - (c + di) = (a - c) + (b - d)i

    (a + bi)*(c + di) = (ac - bd) + (bc + ad)i

    (a + bi)/(c + di) = (ac + bd)/(c2 + d2) + (bc - ad)i/(c2 + d2)

    Design and implement a class named Complex for representing complex numbers and the methods add, subtract, multiply, and divide for performing complex number operations, and override the toString() method to return a string representation for a complex number. The toString method returns (a + bi) as a string. If b is 0, it simply returns a. Implement three constructors: Complex(a, b), Complex(a), and Complex(). Complex() creates a Complex object for number 0 and Complex(a) creates a Complex object with 0 for b. You should also implement getRealPart() and getImaginaryPart() methods for returning the real and imaginary part of the complex number, respectively. Write a test program that prompts the user to enter two complex numbers and displays the result of their addition, subtraction, multiplication, and division.

2. Define a GroceryItem class that represents a request to purchase a particular item at the grocery store in a given quantity (for example, two boxes of cereal). A GroceryItem object should store an item name (a String), an item quantity (an int), and a unit price (a double). Your GroceryItem class should contain the following methods:

    * A public constructor that takes an item name, quantity, and unit price, and assigns them to the appropriate instance variables.

    * A getCost() method that returns the total cost of this item (its unit price times its quantity). For example, three cans of soup that cost 1.50 each would have a total cost of 4.50.

    * A setQuantity() method that sets the GroceryItem's quantity to the supplied value. You

may assume that the argument is always greater than 0.

Next, define a GroceryList class that represents a list of items to buy from the grocery store. Your GroceryList should contain an array that holds GroceryItem objects, along with a variable that tracks the current number of items in the list. You may assume that the GroceryList will never contain more than 10 items. The GroceryList class should have the following methods:

* A public constructor that creates a new, empty grocery list.

* An add() method that takes a GroceryItem object as its argument. This method adds this item to this list if the list has fewer than 10 items.

* A getTotalCost() method that adds up and returns the total cost of every item in the list.

Write a driver program (either a new class, or a main() method in your GroceryList class) that creates a new GroceryList, populates it with several GroceryItems, and then prints the total cost of the list contents.

Grading Guidelines: This lab is graded on a scale of 0-3 points, assigned as follows:

0 points: Student is absent or does not appear to have completed any work for the lab

1 point: Student has written only one program, but it does not compile or run at all due to errors.

2 points: Student has written (or attempted to write) both programs, but only one compiles and runs without error.

3 points: Student has written both programs, and they both compile and run correctly, without any apparent errors.