Java Exam 5: File I/O and Arrays January 29, 2010 Name _____ Period ___

A. 1-D Arrays

The array below is called Mockingbird.

Boo Radley	Dill	Miss Maudie	Scout	Atticus	Tom Robinson	Heck Tate

1. Write the code needed to declare the array and give it size, but not the values shown above.

   _____

2. Using the code from your answer to #1, what "values" would be in each slot of your array? _____

3. Write code to assign "Heck Tate" to the last element of the array, using the length attribute.

   _____

4. Fill in the blanks to indicate what would be printed to the terminal window with each of the following lines of code. If an error occurs, please explain it/indicate what kind of error.

   System.out.println(Mockingbird[1]); // prints _____

   System.out.println(Mockingbird[Mockingbird.length – 3]); // prints _____

B. 2-D Arrays

The array below is called OddMults.

1	2	3
3	6	9
5	10	15
7	14	21
9	18	27

1. Write the code needed to declare the array and give it size, but not the values shown above.

   _____

2. Write one set of nested loops that would assign values to your array. You can use if statements or cases if you don't see the simple pattern of odd multiples here.

3. Fill in the blanks to indicate what would be printed to the terminal window with each of the following lines of code. If an error occurs, please explain it/indicate what kind of error.

   System.out.println(OddMults[0][2]); // prints _____

   System.out.println(OddMults[2].length); // prints _____

   System.out.println(OddMults[3][3]); // prints _____

C. File I/O

1. File I/O has several differences relative to the I/O from a keyboard. Complete the chart below to indicate two differences between these two kinds of I/O. Two examples are written for you – please do not use these as your own.

	In Text I/O…	In Keyboard I/O…
Example 1	The input comes from a file	The input comes from the keyboard
Example 2	Output can append a file or it can overwrite or make a new file.	Output is always appended to the terminal window's contents.
Example 3 (fill in)		

2. Try…catch blocks of code are used in File I/O (they could also be used in Keyboard I/O). What are these lines of code for – be specific about <u>one</u> example of what you are trying to "catch."

D. Program – Arrays and File I/O

Complete a program that will read a text file of CD orders (Orders.txt) then create a receipt as a new outputted text file (Receipt.txt).

- Use a 2-D array (calcArray) to determine tax rate and shipping costs. The array calc array has three columns; they are, in order, a state (1=California, 2=Oregon, 3 = Washington), tax rate, and shipping cost.
- The method for getting the last "word" of any line of text and converting it to a decimal value so a running total can be calculated is written for you (getMoneyAndConvert()).
- One example of Orders.txt is shown below. Contents of Orders.txt can vary, but the first word is always a state to ship to and tax, and the money value for each item is always part of the last "word" of any line beginning with the second line of the file.
- Complete the program written for you, over the next two pages. Comments are provided to help you; when terms are bold, they refer to variable names.

calcArray:

1	0.0825	5.95
2	0	7.95
3	0.065	9.95

Orders.txt:

California
Jack Johnson CD $19.99
Zee Avi CD $14.95
Led Zeppelin CD $9.79
GoGos CD $7.99

For the example Orders.txt above, here is the outputted Receipt.txt:

Jack Johnson CD $19.99
Zee Avi CD $14.95
Led Zeppelin CD $9.79
GoGos CD $7.99
Subtotal $52.72
Shipping $5.95
Tax $4.35
Total $63.02

```
import java.io.*;
import  java.util.Scanner;

public class CDSales{

        double total, ship, tax = 0.0;
        Scanner s = null;
        PrintWriter pw = null;
        String lineIn=null;
        double [] []  calcArray = {{1, 0.0825, 5.95}, {2, 0, 7.95}, {3, 0.065, 9.95}};

                                                                    // make additional variables as needed

public static void main (String [] args){

                                                  //multiple lines: instantiate objects for Text I/O using error checking (they are declared above)









String firstWord =_____; //read first word. Save it as firstWord.  This is used to determine tax and shipping later.

                                                                    //open loop for reading and writing

                                                                    //read Orders.txt line by line, save as variable lineIn

                                                                    //write the line read to Receipt.txt

total = total + getMoneyAndConvert();       //call getMoneyAndConvert() to calculate running total (saved as total).  This line is complete.

                                                                    //close loop

                                                                    //write running total to Receipt.txt as "subtotal"

calculate (firstWord);                                              //calculate shipping and tax.  This line is complete.

                                                                    //write shipping to Receipt.txt

                                                                    //write tax to Receipt.txt

                                                          //calculate final total (add shipping and tax to total) and write it to Receipt.txt

                                                                    //save Receipt.txt


}
```

```
public static double getMoneyAndConvert(){                    //this method is  complete
        double cash = 0.0;
        cash=Double.parseDouble(lineIn.substring(lineIn.indexOf('$')+1));
        return cash;
}

public static void calculateTax(String input){
/*Check what tax and shipping will be by using calcArray and first word in Order.txt, and assign these values to tax and ship variables.  Assume
California=1, Oregon = 2, Washington = 3 in array.  Somewhere in here, calculate tax by multiplying the tax rate times total (don't tax the shipping). */









}

}
```