

IST 256
Lab
Week 7 – Thursday, February 27, 2014

1. Writing a method

Suppose that we have the following code that is used to compute the price of books and add in sales tax.

```
int numBooks = 5;
double pricePerBook = 20.00;
double taxRate = .08;
double bookCost;

bookCost = pricePerBook * numBooks;
bookCost = bookCost + (bookCost * taxRate);
```

Design and write a method definition that will compute the cost of any number of items at any price and with any amount of sales tax, and return the final cost of the items with tax.

Write a method call to compute the price of books, given the original variables.

2. Understanding arrays

Write a piece of a program that will declare an array called *numbers* that can have 25 elements and then has a loop that for every array index i will assign the value $(i * 5)$ to that element of the array.

3. Write a program with exception handling for NumberFormatExceptions

Although Java does not require our programs to handle NumberFormatExceptions and it is not typically done in most programs, we will write a program to handle these exceptions as a way to practice writing try/catch blocks.

In this lab, we will write a simple program that generates values that are multiples of a number. The user will give two inputs:

- the number of values that they want to see (an int)
- the number that is to be the multiplier (a double)

Suppose that the user types in a number n for the number of values and a number m for the multiplier. When the user clicks the button to “Display Multiples”, the program will display a single line result that will have the result of multiplying the numbers from 0 to $(n - 1)$ times the multiplier m .

For example, if the user types in 4 for the number of values and 2.5 for the multiplier, the result will be:

0.0 2.5 5.0 7.5

The program should use an array to hold the numbers for the result, and the program will use NumberFormat Exceptions to make sure that the user inputs valid numbers.

a. Here is a sketch of the user form:

Showing Multiples

Enter the number of values (int) | _____ | (textfield)
Enter the multiplier (decimal) | _____ | (textfield)

|__ Display Multiples __| (button)

Results: (label)

Create the Event -> Action -> actionPerformed method for the button.

b. Write the actionPerformed method for this button:

i. In the code for the button actionPerformed method, declare an integer and a double variable. Pick good variable names for these two variables.

ii. Put a try/catch statement into the code. Since we are writing a program with a GUI, we will include a popup message dialog box to show the error to the user. The try/catch block will have all of the code for the button, except for the initial variable declarations. This is so that if an input error occurs, the entire rest of the button code is skipped.

It will initially have the form:

```
try
{

}
catch (NumberFormatException e)
{
    // print out the Exception
    System.out.println(e.toString());
    // if a number error occurs, show message and return to the user
    javax.swing.JOptionPane.showMessageDialog(null,
        "Please enter valid numbers");
}
```

iii. Inside the try block, put the code to read the two textfields into the declared variables, converting the first one to an integer and the second one to a double.

We will continue to put all the button code inside the try block because we want to skip the rest of the button and return to the user to let them try again.

iv. Add code to declare an array whose size is the integer number. Then write a for loop that for every array index i will assign the value ($i * \text{multiplier}$) to that element of the array. (Refer to problem 2 in this assignment.)

v. Now initialize an output String variable to the empty string and write a loop that will go over every element of the array and append an array value and a space to the output String. For example, if the number of elements is 5, and the multiplier is 3.0, we could create the String:

0.0 3.0 6.0 9.0 12.0 15.0

vi. Display the string in the results label.

d. Test this program.

Try several ways to type in numbers, letters or special characters that will not be either a valid integer or double. Note the message in the output pane as well as the dialog box.

Show examples that you tried that gave NumberFormatExceptions:

Examples:

Hand in your lab sheet and your program by next Tuesday March 4.