# *Programming II (420-B20-HR)*

# *Lab 1 – Java Review*

Date assigned: Wednesday, January 20, 2016

Date due: **Wednesday, January 20, 2016**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

1. Identify and correct Java syntax errors.
2. Review the structured constructs of sequence, selection and iteration in Java.
3. Review screen input and output in Java.
4. Review reading from data files in Java.
5. Review Java frames and event-driven programming in Java.

**To Be Handed In:**

1. The ***username*\_B20\_L01\_Java\_Review** folder should be zipped and uploaded to Moodle.

All java files must be formatted using the **Eclipse** **format** command.

**Getting Started:**

## Create a Course Folder for this course:

### Create a **420-B20** folder on your **H:** drive.

### Create a folder called **Labs** in your new **420-B20** folder.

## Log into **Moodle** and register in **Programming II**. The enrolment key is **420B20**.

## Download the **B20\_L01\_Project** zipped file from **Moodle** and unzip it into your **420-B20\Labs** folder. Rename the unzipped folder to ***username*\_B20\_L01\_Java\_Review**.

## Download **HeritageFormat.xml** from **Moodle** into your **420-B20** folder.

## Start **Eclipse** and select your **420-B20\Labs** folder as your workspace.

## Create a new **Java Project** called ***username*\_B20\_L01\_Java\_Review**.

## Set the formatter to use the Heritage format style:

### Click on the project name in the Navigator and select **Project**🡪 **Properties**🡪**Java Code Style**🡪**Formatter**.

### Click **Configure Workspace Settings…**

### Click **Import…**

### Select **HeritageFormat.xml** from your **420-B20** folder. Click **OK** until you are back at the Eclipse work space.

# Review of Structured Program Constructs

***Purpose:*** Practice using the Java **for**, **while** and **if** constructs

***To Do:***

The **Lab1A** program drills a user on a multiplication table. The user enters a starting number and the program prompts the user for the answer to each of the multiplication problems for that table.

## Open the **MathDrill** class. Correct the syntax errors.

## Open the **Lab1A** class. Correct the syntax errors.

## Run the program. Test it for any integer. Enter a wrong answer to see what happens.

## Modify the program to drill for the table from 1 to 12 instead of from 1 to 10. Test your changes.

## Add a **while** loop or a **do-while** loop to the **multiplicationDrill** method in the **MathDrill** class to reprompt the user for the answer until the correct answer is entered. A partial sample run follows:

What is 5 x 2? 10

Congratulations - correct answer

What is 5 x 3? 12

Sorry - wrong answer. Try again. What is 5 x 3? 13

Sorry - wrong answer. Try again. What is 5 x 3? 17

Sorry - wrong answer. Try again. What is 5 x 3? 15

Congratulations - correct answer

What is 5 x 4? 20

Congratulations - correct answer

## Add a method called **additionDrill** to the **MathDrill** class to test the addition table for the base number.

## In the **main** method of the **Lab1A** class, ask the user whether he/she wants to do an addition drill or a multiplication drill and execute the corresponding method of the **MathDrill** class. Use a switch statement to determine which drill to do. A partial sample run should look like:

Select one of the following:

1. Addition Drill

2. Multiplication Drill

1

What number do you want to test? 5

What is 5 + 1? 6

Congratulations - correct answer.

What is 5 + 2?

## Test your changes.

## Format both classes. (Right-click and select **Source**🡪**Format**.)

# Review of Strings and Data File Reading

***Purpose:*** Review JFrame components and the JFrame event handler.

***To Do:***

## Open **TextCounter.java** in the **Lab1B** package. Run it.

## Modify the program so that instead of reading the sentence from the keyboard, it reads it from the file "my\_favourite\_sentences.txt". Test your changes.

## The file "my\_favourite\_sentences.txt" has several sentences. Modify your program to read all the lines in the file. Display each sentence after it is read. Your final output for the first two sentences should look similar to:

The sentence is "A rat in the house might eat the ice cream."

The words begin at positions 0, 2, 6, 9, 13, 19, 25, 29, 33, 37

There are 10 words in the sentence.

The sentence is "Eighteen bytes is a bit gross."

The words begin at positions 0, 9, 15, 18, 20, 24

There are 7 words in the sentence.

# Review of JFrames

***Purpose:*** Review JFrame components and the JFrame event handler.

***To Do:***

## Open **Lab1CFrame.java** in the **Lab1C** package. Remove the comments from the statements in the **actionPerformed()** method. (You can quickly remove the comments by highlighting the block and clicking **ctrl-/** )

## Correct the syntax errors.

## Run **Lab1CFrame.java** . The frame is supposed to display all the odd numbers between 1 and the value entered in the maximum number text field when the **oddBtn** is clicked.

## Add code to the **displayOdds** method of the **List** class to display all the odd numbers between 1 and **max** in the **display** TextArea.

## Run **Lab1CFrame**. Correct any errors. Try listing the odd numbers between 1 and 15 and between 1 and 20. Try 150. What happens? \_\_\_\_\_\_\_\_\_\_\_\_\_\_

## To automatically wrap the output to the next line, add the following statements to the constructor:

display.setLineWrap(**true**);

display.setWrapStyleWord(**true**);

## Add a second button to the **JFrame** called **btnEven** with the label "*Even Numbers*".

## Add a method called **displayEvens(TextArea display)** to the **List** class which displays all the even numbers between 2 and max.

## Add code to the **actionPerformed** method of the **Lab1CFrame** class to call **displayEvens()** if the **evenBtn** is clicked and calls **displayOdds()** if the **oddBtn** is clicked. Test your changes.

# Homework

## Complete the **Week 1 Quiz** on Moodle by Jan. 24.

Don’t forget to zip and upload your ***username*\_B20\_L01\_Java\_Review** folder to Moodle!