Laboratory 5

CS 1323, Spring 2015

# Learning Objectives

1. Practice hand writing first program. (40 points)
2. Practice hand writing second program. (40 points)
3. Create a log sheet of opportunities for improvement. (20 points)

Section 10: This laboratory assignment is done without a partner. Have the TAs check your work if you finish during lab. If not, upload the two .java files and the log of improvements (in a PDF file) in a single compressed (zip) file.

Section 1: Upload the two .java files and the log of improvements (in a PDF file) in a single compressed (zip) file.

# Description

Students sometimes report having difficulty writing code by hand on the midterm examinations. This laboratory is designed to give you practice with this important step. On the next two pages are problems that are designed to be similar to the programming problem on the midterm.

For each problem do the following steps:

1. Write the code by hand for the first problem without referencing textbooks, the Java API, eclipse, etc.
2. After you have finished, enter your code into eclipse exactly as it was in the hand written version.
3. If your code doesn’t run perfectly immediately, keep a log (see next sheet) of mistakes in your code that you had to correct using eclipse. This will give you a list of topics to study in preparation for the exam.
4. Repeat this process for the second problem.

|  |  |
| --- | --- |
| **Problem #**  **(1 or 2)** | **Description of Problem** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Problem #1

Sitting too much has been linked to health problems. To help people understand how much time they spend sitting, you will write a program that lets the user enter the number of times they stood up during each 30 minute time period. The program will then calculate the number of times the person stood up during the day and the average amount of time they spent sitting without standing up. To make time entry easier, we will represent it as a decimal using a 24 hour clock. So instead of entering 8:30 a.m., you would enter 8.50. Instead of 5 p.m., you would enter 17.0.

The interaction for the program is below. Italics are output from the program, bold is input from the user.

*How many times did you get up between 8.00 and 8.50?*

**0**

*How many times did you get up between 8.50 and 9.00?*

**1**

….13 more entries like that….

*How many times did you get up between 16.50 and 17.00?*

**1**

*You got up 6 times today.*

*Your average time sitting between standing was: 1.5 hours.*

# Problem #2

A restaurant owner has asked you to write a program to keep track of the number of meals that are served during a busy lunch hour. His restaurant serves only 3 items: a special, a hamburger, and a salad. Your program should let the server enter the number and item, and produce a count of what was ordered during lunch. The program should stop when the user enters quit. You may assume that they always sell at least one lunch

A sample interaction is below (italics come from program, bold from user:

*Enter the type (special, salad, or hamburger) of entrée followed by the number, or quit to exit the program.*

**hamburger 1**

*Enter the type (special, salad, or hamburger) of entrée followed by the number, or quit to exit the program.*

**salad 2**

*Enter the type (special, salad, or hamburger) of entrée followed by the number, or quit to exit the program.*

**special 4**

*Enter the type (special, salad, or hamburger) of entrée followed by the number, or quit to exit the program.*

**hamburger 2**

*Enter the type (special, salad, or hamburger) of entrée followed by the number, or quit to exit the program.*

**special 3**

*Enter the type (special, salad, or hamburger) of entrée followed by the number, or quit to exit the program.*

**quit**

*You sold 3 hamburgers, 2 salads and 7 specials.*