Java & OO SE 2

Laboratory Sheet 2

This Lab Sheet contains material based on Lectures 3 – 4.

The deadline for Moodle submission of the lab exercise is 24 hours after the end of your scheduled laboratory session in week 4 *(6 – 10 October 2014).*

You may submit work that is incorrect or incomplete. In order to stretch the stronger members of the class, some of the exercises are quite challenging so don't worry if you can't complete all of them. You should spend around 3 hours per week on programming exercises.

Beginners – start at 'Set Up' Section on page 1.

Experts – skip to 'Submission material' Section on page 2.

Aims and objectives

- Reading Java documentation for standard library classes and using library methods.
- Processing Strings and characters.

Set Up

When you download Laboratory2.zip from moodle, please unzip this file. You will obtain a subfolder Laboratory2. One sub-folder for the projects PalindromeFinder will be created, and the appropriate Java source code file will be in this folder.

Launch Eclipse in the same way as last week. Switch your workspace folder to Laboratory2. Create a new project named PalindromeFinder

In the *Package Explorer view* (on the left of the Eclipse window), find the single source code file for this program, i.e. Palindromes.java. Expanding the default package icon will reveal this file. Then double click the file name to open the file in the central *Editor view*.

The program here is incomplete – you should be able to run it as a Java application – when you run it, it will wait for you to enter some input text on the console - however the isPalindrome() method always returns false. So the output should be something like:

```
Foo

is not a palindrome
```

Submission material

Preparatory work for these programming exercises, prior to your scheduled lab session, is expected and essential to enable you to submit satisfactory solutions.

Unit Tests: I have supplied some simple test cases to check that your palindrome code is working properly. When you create the projects, you should see *two* source code files – one is the skeleton Java source code for you to fill in the details. The other file contains the Unit tests to check your programs are working properly. In the (likely) event that Eclipse complains about your Unit tests, do the following:

In the package explorer pane (top left):
Right click on package > build path > configure build path > libraries > add Library > JUnit. Click Finish.
Now all the Eclipse build errors should go away.

To execute the program:
right click on Palindromes -> run as Java application
right click on PalindromesTest -> run as JUnit test

Submission 1

Edit the isPalindrome() method so it returns true when its input string is a palindrome, and false otherwise. A palindrome is a string that has the same sequence of characters when read forwards as when read backwards, e.g. ABBA, HANNAH, OXO. Note that the empty string is also a palindrome.

Input to the program should come from the keyboard. This is already set up in the main method, which you will not need to change.

Hint 1: Check out the documentation for the String class online – see http://docs.oracle.com/javase/7/docs/api/java/lang/String.html - you need to find a method that selects a single character from a specified position in the String.

Hint 2: Use a `for` loop. You need two variables – one that starts at the left-hand end of the String going forwards, the other starts at the right-hand end of the String going backwards. You need to compare characters pair-by-pair until you get to the middle of the String.

Submission 2 (more challenging)

Some palindromes are multiple words, e.g. "Never odd or even". In these cases, space characters and upper/lower case differences are ignored. Edit the preprocess() method so that it returns a string only containing lower case letter characters – i.e. it strips out spaces and punctuation characters, and converts upper case letters to lower case. The test cases will show you which characters you will need to strip out.

Hint 1. You might iterate through the String and discard characters that are not alphabetical, storing the alphabetical characters into a StringBuffer.

Hint 2. Alternatively you might want to use a regular expression pattern – see the String documentation.

Hint 3. There is a built-in String method to convert all characters to lower case – check the documentation.

How to submit

Please submit your work before the deadline no matter whether the programs are fully working or not.

When you are ready to submit, go to the JOOSE2 moodle site. Click on Laboratory 2 Submission. Click 'Add Submission'. Open Windows Explorer and browse to the folder that contains your Java source code …\Laboratory2\PalindromeFinder\ and drag only the *single* Java file Palindromes.java into the drag-n-drop area on the moodle submission page. **Your markers only want to read your java file, not your class file.** Then click the blue save changes button. Then click submit assignment and fill in the non-plagiarism declaration. Your tutor will inspect your files and return feedback to you via moodle.

Outline Mark Scheme

Your tutor will mark your work and return you a score in the range A1-H. Example scores might be:

A1: you complete both methods correctly and pass all the unit tests
B1: you complete the isPalindrome() method correctly and pass all the single word unit tests
C1: you attempt either or both methods and show that you are trying to implement a sensible algorithm, even if it does not work properly.