Assignment 4: Unit Testing

Weight: 7.5%
Due: Friday, April 17, 2009 at 11:59pm

Assignment Goals

The purpose of this assignment is to give you practice with the concepts of unit tests, Java I/O, design patterns, and HCI. This is not a programming assignment, but you are required to write and modify small blocks of code.

Questions

1. Perform the following sequence of steps in the Student Centre on my.ucalgary.ca (Screenshots available online):
    a. From the menu on the left, select Self Service -> Student Center
    b. From the same menu, select My Favorites -> Add to Favorites
    c. Click OK
    d. From the menu, select My Favorites -> Edit Favorites
    e. Click Delete next to the favourite that you just added
    f. Click OK
    g. Click Self Service in the menu on the left
    h. Read the dialog box
    i. Click OK
    j. Click Save
    k. Sign out of PeopleSoft

Comment on the *usefulness* and *usability* of the ability to add, edit, and remove favourites in PeopleSoft.

2. Write three unit tests using JUnit for the "multiply" method in the Matrix class provided with this assignment. Your tests should include the following:
    a. A normal case of multiplying a n × m matrix with a m × p matrix (which should yield a n × p matrix)
    b. A special case of multiplying a square matrix (n × n) with its identity.
    c. An exceptional case of attempting to multiply a n × m matrix with a p × q matrix (where $m \neq p$)

3. Java I/O:
    a. Modify the following code to make it possible to write and read a tag cloud to and from a file:

```
public class TagCloud
{
    // ... instance variables

    // ... methods
}
```

    b. Modify the following methods so that they can be used to store and retrieve a tag cloud object from a file on the hard drive:

```
public static void writeTagCloud(String filename, TagCloud tagCloud)
{
    // ...
}

public static TagCloud readTagCloud(String filename)
{
    // ...
}
```

4. In Java, it is possible to create what is called an *anonymous class* by overriding a method upon creation of the object. The anonymous class is a subclass of the type that you are creating (with no name of its own). Here is an example of using an anonymous class:

```
JButton button = new JButton("OK");
button.addActionListener(new ActionListener()
{
    // The actionPerformed method in this anonymous class is overridden
    @Override
    public void actionPerformed(ActionEvent event)
    {
        JOptionPane.showMessageDialog(null, "You pressed OK");
    }
});
```

Draw the class model for this example (hint: it should only have two classes) and describe any constraints on the navigability to and from the anonymous class. What is different between this class and the *singleton* design pattern?

Evaluation

Your mark for each part will be calculated as follows:

	Excellent	Satisfactory	Unsatisfactory
Question 1	(25 marks) You have clearly demonstrated that you understand usefulness and usability by stating several reasons that are relevant to their definitions.	(10-24 marks) You have stated only one or two reasons why the program is useful/not useful or usable/not usable and/or your reasons are not appropriate for their definitions.	(0-9 marks) You have stated a reason for only one of either usability or usefulness or you have misunderstood the definitions in the reasons you have given.
Question 2	(25 marks) The TA is able to run all three unit tests on the provided Matrix class and have them pass successfully. The unit tests test exactly what has been requested.	(10-24 marks) The TA is able to run one or two of the unit tests or one or two of them fail when they should have succeeded. One or two of the unit tests may not have tested the requested aspect of the Matrix class.	(0-9 marks) The unit tests do not complete as they should or the unit tests are not testing the correct aspect of the Matrix class.
Question 3	(25 marks) The modifications that you made allow the TagCloud class to be saved to a file, use a minimal amount of code and make use of the correct Java I/O classes.	(10-24 marks) The modifications you made could be adjusted in up to four places to make the code work as expected. The modifications may have used unnecessary code or classes from the Java API.	(0-9 marks) The modifications made do not allow TagCloud objects to be saved to or read from a file, are overly complex, or do not make use of the appropriate Java I/O classes.
Question 4	(25 marks) Your class model accurately represents the given code, your description of its navigability includes all relevant details, and you clearly distinguish between anonymous classes and singleton classes.	(10-24 marks) Your class model is accurate, with the exception of up to one error, but your description of navigability is missing one or two important details. Your distinction between anonymous classes and singletons may have been incorrect.	(0-9 marks) Your class model has more than one error, your description of navigability is incorrect, and you could not distinguish between anonymous classes and singletons.

The TA may deduct up to 5% from the assignment's final mark for errors in spelling and grammar.

Handing in your assignment

For this assignment, email your program and your write-up to your TA on or before the due date. Be sure to check that the format of your final report is one which your TA can read. Make sure that your email client program saves a copy of the email you send to your TA. In the event of email problems, we need the header information from your original email to ensure that you submitted your assignment on time. If you would prefer to hand in a written copy, please hand the assignment in on Thursday, April 16 in class or tutorial, or make special arrangements to meet with the instructor or TA in person before 4pm on Friday, April 17.