

# Programming Lab in Medical Informatics III

BIOMED 223C, University of California, Los Angeles

Short Programming Assignment

Jason Corso

**Given:** April 4, 2006

**Due:** April 21, 2006

**Assignment Goal:** The purpose of the assignment is to refresh your Java knowledge and give you exposure to practical OO design on a small scale. You will use the code written for this assignment as the “front-end” for the quarter project.

**Disclaimer:** Do not worry if you do not understand some terminology or concepts in the assignment description. You will learn them in the coming lectures.

**Overview:** You will write a Swing-based program that will serve as the *pattern classification workbench*. The program will handle loading in an entire dataset from the disk, displaying it, and allowing the user to work with it, which will be better defined below.

**Grading:** 20% compiles, 20% each of the three requirements, and 20% documentation. Please submit by email.

---

As we discussed in class, the application-focus of the quarter is on a pattern classification system. The system will be able to classify a sequence of points into one of a predefined set of classes. The sequence of points can be an open or closed curve, and the dataset will be labeled (i.e., supervised learning).

In this programming assignment, you will write both the low-level code to load the data into memory as well as the user interface for displaying the different datasets. The interface code will be extended throughout the quarter as new functionality (i.e., learning and classification) is added to the codebase.

## Requirements

1. The dataset consists of a set of XML encoded files where each file is a collection of data items of a given class. For example, in file “horse.xml” are numerous contours of horses. The class name is encoded in the xml as well. Your code must automatically get a list of all the xml files in a user-specified directory and then load the data for each class. The xml schema for the data files is as follows:

```
<examples>
<gesture name="gesture-name"/>
<trainingExample>
  <coord x="380.0" y="272.0" t="1016654449468"/>
  . . .
</trainingExample>
. . .
</gesture>
```

Each data coordinate (“coord”) has an  $(x, y)$  pair for spatial location, and  $t$  for temporal information. Note the spatial data is floating point while the temporal data is a long integer. There is only one so-called “gesture” per file. We will cover a JAVA DOM XML parsing example in the lab section of the course.

2. Create a user interface for interacting with the data. It should be able to
  - (a) Display all data classes by their “name.”
  - (b) Allow the user to select a single class and display all of the data items for that class. You can be creative here; either display the items by some unique numerical index, create some thumbnail depiction, both, or some other method.
  - (c) Allow the user to select a single data item from this class. The selection should then display the item inside of a custom JComponent sub-class that you will write (next requirement).
3. The data item display component will be the primary way the user will look at the data items. It must have the following functionality:
  - (a) Scale the canvas to view the entire contour at once and display the scale.
  - (b) Display both the data points and the connecting contour.
  - (c) Give the option to render a grid beneath the contour.

## Main Use Case

Since the system is quite simple at this time, the usability (and utility) is quite minimal. The single use case describing how the system will work is here:

### Main Success Scenario:

1. User loads a dataset.
2. User selects a data class.
3. User selects a data item.
4. User exits.

### Variations

- 1b. Dataset cannot be loaded properly. Error code displayed.

## Suggestions

I suggest you spend time thinking about the structure, or *design*, of your software. Remember that it will become the base of the quarter project. However, it is not expected that you make perfect designs at this point since we will only have begun covering about OO design and analysis. The assignment is intended to get you thinking about OO design. We will *refactor* this code together.

## Minutia

1. You will use the Java 5.0 code level.
2. You will use Eclipse.
3. You will thoroughly document the code in Javadoc. We will discuss Javadoc syntax in the lab section.
4. You will place all of your code in a package named `mi.course.223c.YOURNAME`.

**Good Luck!**