

# **Project 1: Java Applets & XML**

## **Spring 2016**

### **Due Wednesday, April 19, 2016**

#### **Overview**

In this first project you will exploit Java's universal "write-once-run-anywhere" feature by using a Java applet for your Library Checkout System (LCS) Project. Applets are Java's Web-based, client-side application technology, capable of running in any Java-enabled browser. The Java applet will run in a Java-enabled browser on your local computer.

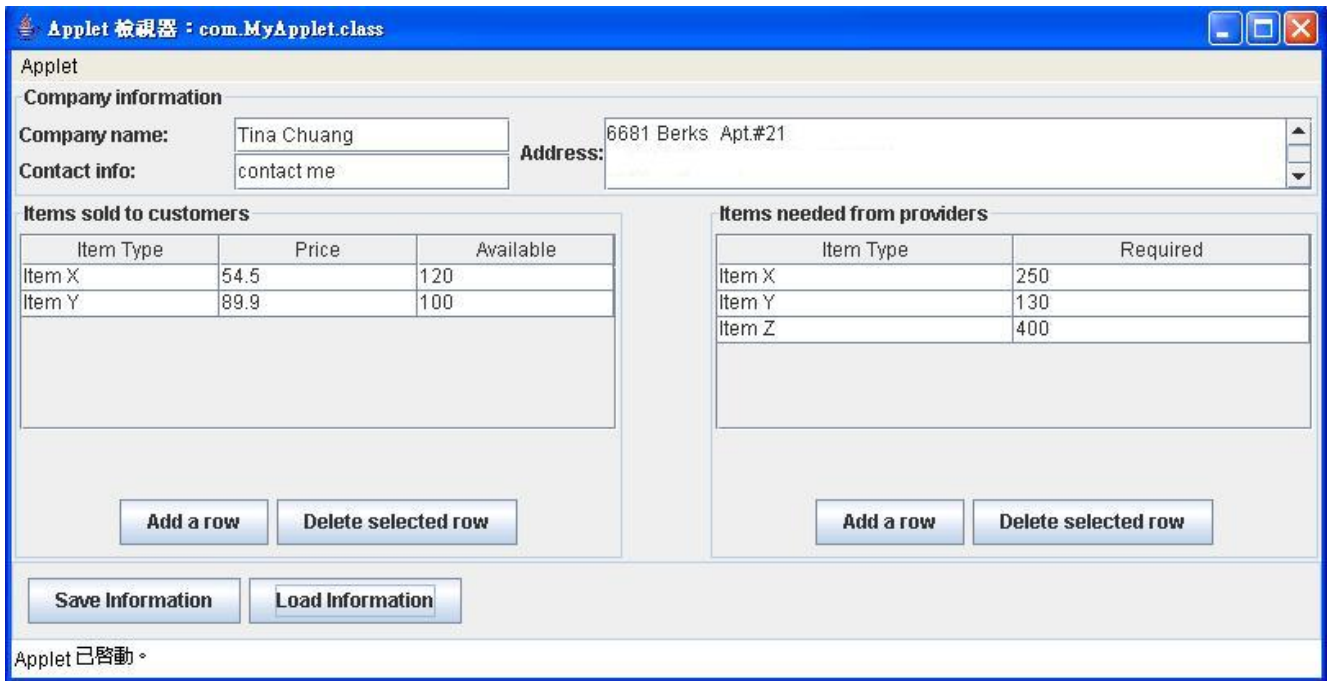
The first project requires you to implement the manufacture part of the Library Checkout System, using a Java applet to record the information about a reader. Your project should contain an applet that has a menu-based Graphical User Interface (GUI), which can allow the reader to insert, update, and delete the information about a reader. In addition, your project should let the reader to store all of its entered information from the GUI to an XML-formatted file after he/she closes the applet. Furthermore, your project should allow the reader to load the XML-formatted file to the GUI, which would automatically loads all the contained information from the XML-formatted file to the GUI. We have provided necessary XML parser classes, as well as the project template for this project on the class Web site.

Your project should contain an applet that has a menu-based Graphical User Interface (GUI) with the functionality listed below:

- Reader information (first name, last name, contact info., keep record of last login, address, etc.)
  - Add information
  - Edit this information
  - View this information
- Library information
  - Add new library(s)
  - Modify existing library(s)
  - Delete existing library(s)
  - View item(s)
- Library Checkout-related Information (checkout/return dates, checkout items, checkout quantities, etc.)
  - Add new information(s)
  - Modify information(s)
  - Delete information(s)
  - View information(s)
- Load information from a selected XML-formatted file
- Save current information to a selected XML-formatted file

In this project, you will be required to implement the reader side. You will reuse most of the code in subsequent projects. Therefore, it is better to put in more time and thought at this stage to avoid more work in later projects.

You may use design your GUI like the following (Please be aware that this is just for the reference from the previous year, but it's not necessary that you need to follow this layout for your project)



## Swing GUI

This assignment requires that you build a GUI using the Java Swing toolkit, which enables you to produce a GUI quite quickly and easily. You should spend some time thinking about the types of GUI elements that you will use before you begin coding (read the overall project description and think about future GUI requirements). While much of your input will be obtained using text input fields, consider the use of other elements such as drop-down lists for standard inputs. You should also consider the overall look-and-feel of your GUI. *We want to see polished GUIs, rather than disorganized GUI elements without a general cohesiveness.*

## XML Format

You will format the information about the reader using the eXtensible Markup Language (XML), so that the data are self-describing and portable across different hardware and software platforms. You will need to implement methods for writing this information in XML format. You will be provided with the necessary parser classes for XML files.

## Design Considerations

The data should be obtained from the user in a clear and easily understood manner. *Prompting for information is a suitable mechanism. In case of errors, inform the user.* Ultimately, the behavior of the system is up to you. The exact information that your system stores is at your discretion, but it should make sense in terms of the project. You are encouraged to think about the project, and to discuss it with the TA and the professor for further understanding.

## Security Issues

Because of the nature of applets, there are several security issues related to an applet's reading and writing files on your local computer. Instructions on how to deal with those issues will be provided on the class Web site and also from our previous lectures. The template from the class Web site includes necessary scripts to get started with the project.

## Plagiarism

TA will be checking programs for plagiarism, so please don't copy from anyone else. If we catch you plagiarize someone's work, both of you will get zero point for this project!

## Report

In addition to the programming portion of this project, you will write a small report. The report should be clear, well organized and concise. It needs to include the following:

- Explanation of how your system works. You can simply provide a lot of screenshots to guide us through your system
- Details of your design. Please briefly discuss about how you design your system (such as why you build your system in this way...)
- A class diagram sketch outlining the hierarchy of your class structure. You can sketch how you organize your files (java, class, build, jar, etc.) in your directory.
- *Sample input and output, e.g., the content of your XML elements before and after an operation, such as add or update.*
- Provide other files as you wish

## README File

You are required to submit a separate README file that explains how to compile and run your program, etc. If you find yourself needing to edit the build.xml file, then you should address what you have modified, as well as providing us the example of executing your build file. If you have encountered any difficulty while you develop this project, feel free to discuss the issue in this file.

## Demonstration

You will be performing project demonstrations for this first project and all others. You will be responsible for scheduling a 5-15 minute timeslot for your demonstration with the TA. Because time is limited, you should run your application on the PCs on which you will do your demo before the demo. In particular, make sure that those PCs are set up correctly before you perform your project demonstration.

Demonstrations will be held on **April 19**, and will fall between specific times. You will be required to demonstrate your application's functionality, and answer several questions covering the technology and your particular implementation. Demonstrations for the first assignment will be scheduled through email, or during TA office hours/discussion.

## Submission

You will submit a compressed file (*rar, tar, or zip*). You will need to type "*ant submission*" in order to create your submission file. If there is anything missing in the submission file, edit the *build.xml* file to include the missing information.

Make sure you modify the *User* property in the *build.xml* file. The *ant* call will create submission file with the username you provide.

The compressed file should include a folder with all project content necessary to build/run the program (.java, README, Report). The project file is to be uploaded to the elearning website, titled **"Project 01: Name LastName"**. Please be aware that you will get **3 points deducted if you don't have correct title** of your project file.

## Grades Distribution:

- Demo: 40%
- Question: 40%
- Codes: 10%
- Submission & Report: 10%
- Extra: up to maximum of 2% for more fancy work.