# CS372 – Java Application Development

## Class Project

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Due: 24 Jan Date Received: \_\_\_\_\_\_\_\_\_

* 0th Deliverable due 16 Jan
* First Deliverable due 18 Jan
* Final Project due 24 Jan

Teams of 2

Be sure to take advantage of the Javadoc comments.

Advice: Too often I hear, “I was up until 5 this morning finishing,” because students procrastinate, and then find that their project isn’t complete. Take advantage of the time we have now. Plan ahead, and schedule for yourself when certain features should be done.

**0th Deliverable (16 Jan):** Email me your application idea. Your description can be vague at this point, but I want evidence that you have been thinking about your project.

**First Deliverable (18 Jan)**: You must come up with your own functional specification. This includes a good description of the project (at least three paragraphs), all class hierarchies (students who have taken CS273 should include UML) and other design documentation. Note, the specification can (and likely will) change over the course of the project. Be sure to document all changes to your specification as it evolves. A rubric will be available soon.

**Final Deliverable (24 Jan):** There are three parts: First, the project itself must be in your github repository. Second, you are required to turn in a write-up of the project, documenting personal experiences in the project. This should include: design and implementation details (especially changes from the functional specification) and a list of concepts used in the project from class. Finally, you will give a class presentation of your work. Rubrics for these will be available soon.

Project:

* You can design your own project, but it must be approved by the instructor before the first deliverable is due (via email).
* Guidelines for your project: Class hierarchy, exception handling, system testing, interface programming where appropriate (roll your own or implement existing Java interfaces), Java features such as generics, text processing, GUI, Android, …
* Extra credit for including one or more of: {Multiple threads, HTTP access, Networking, other concepts you learn outside of class related to Java programming}