OOAD Homework 6 Grading Rubric – Due 4/26 at 11 AM (no late submissions accepted)

Final Report as PDF (50 points) (-50 points if late or missing)

- Name of Project and All Team Members (-5 points if missing in whole or part)
- Documented Final System State
    - Discuss state of system, what features are implemented?
    - List of features implemented (-5 if missing, -2 if not thorough/complete)
- Class Diagram
    - UML showing final set of classes and relationships in the systems (-5 if missing, -2 if not thorough/complete)
- Final Project vs. Initial Design
    - Include Class Diagram from Homework 4 to illustrate changes (-5 if class diagram missing)
    - Discussion of changes (-5 if missing, -2 if not thorough/complete)
    - If no changes from initial design, this should be discussed
- Third-Party Code vs. Original Code
    - What external code was used, what code did you write?
    - List of third party code elements (libraries, utilities) with cited sources (-5 if missing and third party elements used, -5 if sources not cited in list, -2 if not thorough/complete)
    - List of original code elements of design (-5 if missing, -2 if not thorough/complete)
- Design Patterns
    - Were any used, if so how?
    - Even if no patterns were used, this should be stated and discussed.
    - (-5 if missing, -2 if not thorough/complete)
- Learnings on OOAD
    - What has been learned about the process of OO analysis and design from this project?
    - (-5 if missing, -2 if not thorough/complete)

Code and README Submission by Repo URL (50 points) (-50 points if late/missing)

README Markdown or document in Repository (20 points)

- Name of Project and Names of team members (5 points)
- Basic project overview (5 points)
- Descriptions of files in repo (5 points)
- Notes on installing or executing (5 points)

Well commented modular OO code (30 points)

- Code is Object-Oriented in Design
    - (15 points, -5 or -2 for functional or procedural elements that should have been OO)
- Thorough code comments (15 points, -10/-5/-2 if incomplete/unclear)
    - file purpose and authors
    - class or method's purpose
    - key attribute/variable use (not including temporary/counter variables)
    - highlighting pattern use
    - identifying third-party elements (source citations in the code are optional)
    -

Note: I stated in Lecture 23A that "no work should be done on the project after it is submitted". I am loosening this a bit, with the understanding that sometimes preparing for a demonstration uncovers issues. You will be graded on your code submission based on what's turned in on 4/26 before 11 AM. If you decide to make code updates after that point, that is up to you and your team. I encourage you to control the scope of your project work and to try to complete your work by the 26th and not have to do further work to prepare for a demo.

Project Demonstration (50 Points) (-50 if code not demonstrated to Bruce or Manjunath before 5/2)

- Target 15 minutes, as many team members as feasible should attend or join
- Walk-through of primary operational features to be demonstrated
    - (no understanding of features -10, some confusion on features -5)
- Execution/demonstration of submitted code
    - (no operation -10, some major bugs -6, some minor bugs -3)
- What was most challenging issue during development for each team member present
    - (no clear answers -10, poor quality answers -5)
- Team responses to reviewer questions
    - Typical questions
        - Pattern use or OO design considerations
        - Review of libraries, tools and languages used
        - Surprising problems or discoveries
        - Anything particularly hard or easy
        - If you could do it again, what would you do differently
    - (not able to answer questions on project elements -10, poor quality answers -5)