Preparing People to Lead Extraordinary Lives
LOYOLA UNIVERSITY CHICAGO

DEPARTMENT OF
Computer Science

Comp 271 – Capstone Team Project

Purpose:

1. Demonstrate and expand your knowledge of Object Oriented Programming concepts we have learned in class by building a realistic app that builds on past course work. Understand the effectiveness of your OOP class structures and encapsulation as requirements evolve.
2. Work with others to agree on updates as you combine your OOP classes with theirs and integrate your code into a new expanded app.
3. Conduct and evaluate each other's code – can you recognize and produce "beautiful code"?

Logistics:

1. You will be assigned to a team with others from the course.
2. Each team member will work from one part of their past work. Areas for each team member will also be assigned and may be changed only during the initial few days of project work with permission of the instructor.
3. Portions of class time will be allocated for working with your team; attendance is required and will be recorded. Additional work outside of class may also be required; if needed, you must arrange it with the agreement of your team members.
4. See course notes for schedule details; your team will demonstrate and describe your work together to the class.

Technical Requirements:

1. Use the OOP concepts and techniques we have covered in class.
2. UML class diagram(s) required covering and consistent with all code used in the combined system. A UML overview of the entire system is required. Diagrams to be drawn in the tool used in class.
3. Code to be created, tested, and compiled using the tools used in class. You will need to organize the code into a combined project structure that all team members will use.
4. Test data suitable for the work will likely need to be expanded from that used in class. It must show expected results before the tests are run.
5. Your code **must** be readable, well structured, and use the standard Java style used in class, including naming conventions and limitations on size of individual methods.
6. At least some of your testing must be driven by user input and allow for errors in the input (incorrect values, responses, etc). Use simple text based console I/O for input and output.
7. DO NOT: use other online solutions, AI tools, Chegg, etc. All work is to be created by the team members during the course. Violation of this requirement will be reported to the appropriate university college and may result in a failing grade for the course.

What to Turn In:

1. A single Overview UML
2. Multiple UML Class Diagrams covering all of the app, will organized, consistent with each other and the overview UML
3. Full code as separate source files and consistent in all ways with the UML design
4. Results of team code reviews with metrics on changes
5. Presentation materials

Presentation Outline (approximately 10 minutes):

1. Team and app introduction

2. Demo of your working app including GUI use
3. Design approach and challenges in combining your separate work. What did you learn to do more of or differently when making OOP apps in the future?
4. Code review results. What value did you see, if any, in reading other's code?
5. Results of team code reviews with metrics on changes
6. Anything else you wish to cover.

Problems:

Problems with team member's participation, commitment, communication and any other concerns you may have, please share with instructor via email with course title in subject line. Share problems as soon as possible to allow time for corrective actions.

Team members who are unable to fully participate and meet expectation of their team members may receive zero points for the capstone.

Grading rubric:

1. OOP Structure (25 points team score) – effective encapsulation, appropriate coupling and cohesion, appropriate use of containers, correct application of key OOP methods (e.g. equals, compareTo, toString,…)
2. Beautiful Code (25 points team score) – evidence of code reviews between team members; appropriate style and structure of resulting code across all team members.
3. Team Participation (25 points personal score) – your personal contribution to the work with the team including full attendance in participation in team activities in and outside class time.
4. Demonstrated Understanding of App (25 personal score) – evaluated from your contributions to the team materials and presentation.

Team score will be same for all members.

CapStoneTeamProject (original)

WLHonig March 2024 COMP 271 Loyola University Chicago