JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Computer Science 600.123
The Course Title
Semester, Year (3 credits, EQ)

Instructor
Professor Ali Darvish, darvish@jhu.edu, https://cs.jhu.edu/~darvish
Office: Malone 205, 410-516-8775
Office hours: TBA, check course website

Head Teaching Assistant
Steven Tan, wtan12@jhu.edu
Office hours: TBA, check course website

Meetings
Tuesday, Thursday, 1:30 – 2:45pm

Textbook

- Recommended: "Head First Object-Oriented Analysis and Design" by Brett D. McLaughlin, Gary Pollice, Dave West.

Online Resources

- The course website is https://jhu-oose.github.io/cs421_f21/.
- Piazza (piazza.com/jhu/fall2021/en601421621) will serve as our communication channel. You should sign up for the course Piazza page immediately.
- Gradescope (https://www.gradescope.com/courses/297155) will be used for homework assignment submission; entry code is KYGY4D
- Github (https://github.com/jhu-oose/) is used for source code and project management

Course Information

- **Description**
  This course covers object-oriented software construction methodologies and their application. A major component of the course is a large team project on a topic of your choosing. Course topics covered include object-oriented analysis and design, UML, design principles, refactoring, program testing, code repositories, team programming, and code reviews.
- **Prerequisites**
  600/601.226 (Data Structure) and 600.120/601.220 (Intermediate Programming). Students may receive
  credit for only one of 601.421/621.

Course Goals

By the end of the course students will be able to:

- Explain the client-server model and structure a software system that adheres to this software architecture.
- Write a software requirements specification (SRS) document that describes functional requirements of a medium-size software system.
- Priorities functional requirement of a software system according to its requirement specifications.
- Recognize software process models and distinguish between plan-driven models (such as waterfall) versus incremental development approaches (such as agile).
- Follow agile software development process (short iteration, continues delivery, etc.) to build a medium-size software.
- Describe how version control can be used to help collaborative software development.
- Demonstrate the capability to use software tools (Git/GitHub, Wireframing tools, build tools like Gradle, dependency management tools like NPM, IDEs, . . .) in support of the development of a software product of medium size.
- Articulate object-oriented design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation.
- Use UML class diagrams to express the design of a simple software system and explain how system design principles have been applied in this design.
- Design, implement, document and test web application programming interface (web API) for a simple software system.
- Design and implement data persistence strategies (such as databases, cookies and sessions in web applications) for medium-size multiuser software application.
- Describe the SOLID design principles and apply them to then design of small software systems.
- Provide examples of creational, structural and behavioral design patterns in Object-Oriented programming paradigm.
- Within the context of object-oriented paradigm, describe one or more design patterns that could be applicable to the design of a simple software system.
- Refactor an existing software implementation to improve some aspect of its design.
- Write a software component that performs some non-trivial task and is resilient to input and run-time errors

Course Expectations

All students are generally expected to attend all the sessions of this course and actively participate by answering and asking questions. Participation will not be tracked. The main component of this class is a semester-long group project (6-7 students per group.) Each class meeting session will be either a lecture session or a project meeting session following the schedule given in the course website. Students are expected to complete several homework, a quiz and a semester-long project (details are provided below.)

Key Dates

Check course website for complete schedule.

Homework

There are several mandatory homework assignments. These assignments are meant to help with solidifying your understanding of the core concepts and technologies covered in the class. Homework 1 a 2 are completed individually, but the rest are completed in group (your project group). Each group has a total budget of three late days for the homework. Each group may not use more than two late days on any given homework. We will keep track of the late days for each group. Each homework assignment will be assigned a point value; the overall homework assignments grade will be computed as your total points earned divided by the total achievable points. Homework assignments are due at **11pm of the due date.**

Project

A major component of the course is a semester long term project with bi-weekly deliverables (iterations). Each iteration of the project will be evaluated out of 90 points; there will be a final presentation (deliverable) worth 150 points. The overall project grade will be computed as total points earned divided by the total achievable points. The deliverables for each iteration will generally be based on "students plan" for that iteration; i.e., their progress is measured against their plan/goals.

Quiz

There will be a quiz around midpoint in the semester (*tentatively scheduled for Thu 10/19 during the class time*). The quiz focuses on object-oriented design patterns and principles. Students will not be asked to write code. They must, however, be able to e.g. critique a design provided in a UML format. They must also be able to e.g., define (and elaborate on) each design principles/patterns covered in this course. The quiz is individual, timed, closed-book and closed-note, and proctored. You will be provided with the necessary information about the quiz assessment as well as sample prep material once the quiz date is close.

Grading Scheme

- 0% - Class participation (expected, but not tracked)
- 28% - Homework
- 12% - Quiz (individual, timed, proctored, closed-book)
- 60% - Project

Letter grades for the course will be subject to the instructor’s evaluation of your overall class performance, generally based on this standard scale:

> 90	A+, A, A
> 80	B+, B, B-
> 70	C+, C, C-
>= 60	D+, D
< 60	F

The cutoff for assigning -/+ to each grade letter is further subject to the instructor's evaluation of your overall class performance. Do not expect a curve in this course.

Ethics

The strength of the university depends on academic and personal integrity. In this course, you must be honest and truthful, abiding by the *Computer Science Academic Integrity Policy*:

> Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.
>
> Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition. Ignorance of these rules is not an excuse.
>
> Academic honesty is required in all work you submit to be graded. Except where the instructor specifies group work, you must solve all homework and programming assignments without the help of others. For example, you must not look at anyone else's solutions (including program code) to your homework problems. However, you may discuss assignment specifications (not solutions) with others to be sure you understand what is required by the assignment.
>
> *If* your instructor permits using fragments of source code from outside sources, such as your textbook or on-line resources, you must properly cite the source. Not citing it constitutes plagiarism. Similarly, your group projects must list everyone who participated.
>
> Falsifying program output or results is prohibited.
>
> Your instructor is free to override parts of this policy for particular assignments. To protect yourself: (1) Ask the instructor if you are not sure what is permissible. (2) Seek help from the instructor, TA or CAs, as you are always encouraged to do, rather than from other students. (3) Cite any questionable sources of help you may have received.
>
> On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and

dated]". Your course instructors will let you know where to find copies of old exams, if they are available.

Report any violations you witness to the instructor.

You can find more information about university misconduct policies on the web at these sites:

- For undergraduates: http://e-catalog.jhu.edu/undergrad-students/student-life-policies/
- For graduate students: http://e-catalog.jhu.edu/grad-students/graduate-specific-policies/

Personal Wellbeing

- If you are sick, in particular with an illness that may be contagious, notify me by email but do not come to class. Rather, visit the Health and Wellness: 1 East 31 Street, 410-516-8270. See also http://studentaffairs.jhu.edu/student-life/support-and-assistance/absences-from-class/illness-note-policy/
- All students with disabilities who require accommodations for this course should contact me at their earliest convenience to discuss their specific needs. If you have a documented disability, you must be registered with the JHU Office for Student Disability Services (385 Garland Hall; 410-516-4720; http://web.jhu.edu/disabilities/) to receive accommodations.
- If you are struggling with anxiety, stress, depression or other mental health related concerns, please consider visiting the JHU Counseling Center. If you are concerned about a friend, please encourage that person to seek out our services. The Counseling Center is located at 3003 North Charles Street in Suite S-200 and can be reached at 410-516-8278 and online at http://studentaffairs.jhu.edu/counselingcenter/

Classroom Climate

I am committed to creating a classroom environment that values the diversity of experiences and perspectives that all students bring. Everyone here has the right to be treated with dignity and respect. I believe fostering an inclusive climate is important because research and my experience show that students who interact with peers who are different from themselves learn new things and experience tangible educational outcomes. Please join me in creating a welcoming and vibrant classroom climate. Note that you should expect to be challenged intellectually by me, the TAs, and your peers, and at times this may feel uncomfortable. Indeed, it can be helpful to be pushed sometimes in order to learn and grow. But at no time in this learning process should someone be singled out or treated unequally on the basis of any seen or unseen part of their identity.

If you ever have concerns in this course about harassment, discrimination, or any unequal

treatment, or if you seek accommodations or resources, I invite you to share directly with me or the TAs. I promise that we will take your communication seriously and to seek mutually acceptable resolutions and accommodations. Reporting will never impact your course grade. You may also share concerns with the department head (Randal Burns, randal@cs.jhu.edu), the Director of Undergraduate Studies (Joanne Selinski, joanne@cs.jhu.edu)), the Assistant Dean for Diversity and Inclusion (Darlene Saporu, dsaporu@jhu.edu), or the Office of Institutional Equity (oie@jhu.edu). In handling reports, people will protect your privacy as much as possible, but faculty and staff are required to officially report information for some cases (e.g. sexual harassment).