Software Construction and User Interface (SE/ComS 319)

Ali Jannesari

Department of Computer Science

Iowa State University, Fall 2018

ADMINISTRATION AND CLASS INFORMATION

Outline

- Administrative stuff
    - Who I am, TAs, general information, etc.
    - Course organization and syllabus
    - Schedule and structure of this class

About me

- Instructor: Ali Jannesari (Assist. Prof. at CS department)
- My background: PhD at KIT (Germany), RWTH Aachen, TU Darmstadt, UC Berkeley and ISU
- Research: Software engineering, systems, parallelism, deep learning.
- Teaching: I enjoy interactive classes and discussions
- Contact me
  - jannesar@iastate.edu
  - Atasanoff, Room 207, office hours: M 4-5PM
  - Emails to me regarding the class must include "319" in subject line.

TAs

- TAs for Coms 319:
  - Shibbir Ahmed, shibbir@iastate.edu
  - Rangeet Pan, rangeet@iastate.edu
  - Eric Lin, eylin@iastate.edu
  - Subrahmanyam Vaddi, svaddi@iastate.edu

- TAs office hours
  - F 3-4pm Pearson, Room 108

Your background?

- How good are you in programming?
    - Java, C/C++?
    - web programming?
        - HTML, JS, PHP?
- How many programs have you written?
    - Which programing languages do you use?
    - Which IDEs?
    - How big are your programs? 100 LOC? 1k? 5k?
- How familiar are you with OO and UML?
- Are you familiar with agile software development methods (e.g. XP, scrum, etc.)?

General information

- https://canvas.iastate.edu/courses/52828
  - Gives you access to the course material, etc.
- Class schedule
  - Lecture: Monday, Wednesday, Friday 2:10 – 3:0 PM
  - Lab activities and project meetings: Pearson Labs 105/109/108/113/158 for **Friday classes**
- Email communication must start with ”319:" in the subject line
- Prerequisite:
  - COM S 228 (Pre-Req Waiver Form)
  - Knowledge of programming (Java)

Course description

- Basic theory of grammars, parsing. Language paradigms. State transition and table-based software design. Review of principles of object orientation, object oriented analysis using UML. Frameworks and APIs. User interface architecture, evaluation of user interface. Design of windows, menus, and commands. Introduction to formal specification and model-based software design. Introduction to domain-specific software engineering.

Course learning objectives

- Be familiar with web user Interfaces, event-driven programming (client/server, JavaScript, Node.js, PHP, frameworks, etc.)
- Be familiar with software process, OO analysis and UML
- Be familiar with software design principles, design patterns and architectural styles.
- Be introduced to with parallelism and concurrency concepts such as threads and multi-threaded programs
- Be introduced to software testing, test-driven development, agile methods and eXtreme Programming (XP).
- Be introduced to language grammar, lexer and parser

Course overview (tentative)

Week#	MON	Tentative Schedule
Week 1	Mon 8/20	Introduction and basics, threads, client/server programs
Week 2	Mon 8/27	Review of software development process, user interfaces
Week 3	Mon 9/3	Web user interfaces, event-driven programming
Week 4	Mon 9/10	Event-driven programming, web programming
Week 5	Mon 9/17	Event-driven programming, web programming
Week 6	Mon 9/24	Software process, eXtreme Programming (XP – agile methods)
Week 7	Mon 10/1	Review of Object Orientation, OO analysis
Week 8	Mon 10/8	System modeling and UML
Week 9	Mon 10/15	UML diagrams
Week 10	Mon 10/22	Architectural styles
Week 11	Mon 10/29	Design Patterns
Week 12	Mon 11/5	Software Testing
Week 13	Mon 11/12	Parallel Software Construction
Week 14	Mon 11/19	Fall break
Week 15	Mon 11/26	Basic Theory of Grammars and Language Paradigms
Week 16	Mon 12/3	No Class (reading period)
Week 17	Mon 12/10	**Final Exam**

Course outcome (ABET outcome)

- 1. An ability to analyze a complex computing problem, and to apply principles of computing and other relevant disciplines to identify solutions.
- 2. An ability to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
- 5. An ability to function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.
- 6. An ability to apply computer science theory and software development fundamentals to produce computing-based solutions.

What you need to do

- Attending lectures/labs and project meetings **5%**
  - Pearson Labs 105/109/108/113/158 for Friday classes
- Quizzes **10%**
- Homework assignments **20%**
- A main group project (portfolio) **25%**
  - 5% proposal; 5% final presentation and slides; 15% project content
  - Milestones and checkpoints
- Final exam **40%**
- In case you are going to drop this course, do it soon.

Quizzes

- ~10 minutes @ every Wednesdays (Web-based using Canvas during lecture)
- The 1st quiz is on August 29th (Wednesday)
- Close-book, close-note
- Cover lectures and lab activities
- 1 point for each quiz, 10% of your overall grades!

Assignments

- Homework assignments
  - Problems for you to do (individual)
  - Code and explanations (comment your code)
  - **Mandatory**
  - TBA (including report format and submission check list)
  - Selected solutions for code walk, explanation and demonstration during lab sessions

Group project (Portfolio)

- Do it in team (4 students)!
- Grading of group project:
  - Individual performance assessed
- We will look for all of the below:
  - Evidence of vigorous interactions with materials (questions, insights)
  - Exploration of new and complex issues (examples, explanations)
  - Evidence of working at higher levels of blooms taxonomy: analysis, evaluation, synthesis.
  - Evidence of teamwork
- These requirements will be explained later as well.

Final Exam

- Final exam
  - During the exam week
  - 40% of your final grades
- Exams will cover material from class, labs, assignments and projects

Miscellaneous

- A lecture day to be assigned for project works and lab activities @Pearson lab rooms
  - Lab activities and project meetings: Pearson Labs 105/109/108/113/158 for **Friday classes**
- Start early, look for online materials, tutorials on web programming, JavaScript, Node.js, agile software development and XP (eXtreme Programming) process model
- In case you copy/paste any code available in public domain you need to cite it in your source code!
- Slides will be available in PDF on Canvas
  - Only for the purpose of this class. **Redistribution not permitted!**