ECE 448/528 – Application Software Design
Spring 2020

Instructor: Professor Jia Wang
Office: 317 Siegel Hall
Phone: 312-567-3696
E-Mail: jwang@ece.iit.edu (Please start your email subject line with [ECE448].)

Prerequisites: Computer programming. If you haven't been writing programs for a while, please refer to the following book for introductory Java programming.
• "Head First Java", 2nd Ed., K. Sierra et al., O'Reilly Media, 2005. ISBN: 978-0596009205

Reasonable accommodations will be made for students with documented disabilities. In order to receive accommodations, students must obtain a letter of accommodation from the Center for Disability Resources and make an appointment to speak with me as soon as possible. The Center for Disability Resources is located in the Life Sciences Building, room 218, 312-567-5744 or `disabilities@iit.edu`.

Class Time and Location: Mon./Wed.: 11:25 AM – 12:40 PM, Wishnick Hall 113

Class Home Page: `http://www.ece.iit.edu/~jwang/ece448-2020s/`

Required Textbook:
• "Java Network Programming", 4th Ed., E. R. Harold, O'Reilly Media, 2013. ISBN: 978-1449357672
Recommended Textbooks:
• "Core Java Volume I–Fundamentals", 10th Ed. or later, C. S. Horstmann, Prentice Hall.
• "Core Java Volume II–Advanced Features", 10th Ed. or later, C. S. Horstmann, Prentice Hall.
• "Head First Design Patterns", E. Freeman et al., O'Reilly Media, 2004. ISBN: 978-0596007126

Course Summary: This course provides introduction to languages and environments for application software development utilizing Software as a Service (SaaS) for electrical and computer engineers. Student will develop a data-rich web application with server back-end that connects mobile devices and Internet of Things using Agile software engineering practices.

Topics Covered:
• The Java ecosystem.
• Client-server architectures and RESTful services.
• Pub/Sub middleware.
• Web user interface design.
• Security; database; data visualization.
• Software engineering and Agile development.

ECE 448 Grading: Homeworks 10% / Projects: 120% (30% extra).
A: $\geq$ 90% / B: $\geq$ 80% / C: $\geq$ 60% / D (undergraduate only): $\geq$ 55%.

ECE 528 Grading: Homeworks 10% / Projects: 100% (10% extra).
A: $\geq$ 90% / B: $\geq$ 80% / C: $\geq$ 60%.

Homework and Project Policy: Late homeworks and projects will not be graded. Homeworks will be graded based on general approach and completion, and solutions will be released shortly after due date. Discussions on homeworks/projects are encouraged, but copying will call for disciplinary action.

Lecture Schedule (tentative):

No.	Date	Topic	Chapters	HW Out	Project Due
1	1/13, 1/15	Introduction		#1	
2	~~1/20~~, 1/22	Java Overview			
3	1/27, 1/29	TCP/IP and HTTP	1,4-6	#2	
4	2/3, 2/5	Socket Programming I	2,8		Project 1
5	2/10, 2/12	Socket Programming II	3,9	#3	
6	2/17, 2/19	Pub/Sub and MQTT			Project 2
7	2/24, 2/26	Spring Boot Introduction			
8	3/2, 3/4	RESTful Service I		#4	
9	3/9, 3/11	RESTful Service II			Project 3
10	~~3/16~~, ~~3/18~~	**Spring Break**			
11	3/23, 3/25	JavaScript and DOM			
12	3/30, 4/1	Reactjs Introduction		#5	
13	4/6, 4/8	Web UI Design			Project 4
14	4/13, 4/15	Security	6,10		
15	4/20, 4/22	Database Integration			
16	4/27, 4/29	Data Visualization			Project 5
17	5/4–5/8	**No Final Exam**			Project 6

Course Objectives (ABET)
After completing this course, the student should be able to do the following:

1. Understand application software architectures and application software development processes.
2. Utilize event-driven programming to support networking and graphical user interface in application software.
3. Design and implement testable class types. Document and validate functionality via unit testing.
4. Reuse existing class libraries to improve code quality and productivity.
5. Construct reusable class libraries using polymorphism.
6. Utilize design patterns when designing and reusing class libraries.
7. Be familiar with advanced topics including security, database, and data visualization.
8. Design and implement a networked application software with graphical user interface following test-driven and iterative/incremental software engineering practices.