New Jersey Institute of Technology University Heights Newark, NJ 07102
http://njit.edu

CS 673 Software Design and Production Methodology – Fall 2024

Part I: Course and Instructor Information

Semester	Fall 2024
Course	CS 673
Instructor	Prabhat Vaish
Course Meeting	06:00 – 08:50 PM Wed
Office Hours	https://njit.webex.com/meet/pvaish (Links to an external site.) or in the office - Available Tue between 4-6 PM

Part II: Course Description

1. Course description:

This is a course in software engineering, with an emphasis on design and production. It focuses on building web-based applications and services because of their obvious relevance to students and professionals in building practical skills. In addition to teaching the fundamental notions of structuring a network or web application, and how to write modular and elegant code, the course emphasizes behavioral and conceptual design: that is, designing the external behavior of the software rather than its internal structure. I believe that this aspect of design is the most important in practice, the least well-understood by students, and the one that is hardest to acquire in standard industrial settings.

Modern techniques and methods employed in the development of large software systems, including a study of each of the major activities occurring during the lifetime of a software system, from conception to obsolescence and replacement. Topics include cost/performance evaluation, documentation requirements, system design and production techniques, system verification techniques, automated aids to system development, and project organization and management.

For the latest course information go to https://njit.instructure.com/courses/39027

The information below should help you plan and organize your preparation during the semester.

2. Prerequisite courses and knowledge:

- Prerequisite course: None
- Required background:
  - The students are required to have knowledge of key systems concept, software development life cycle, and programming in Java or Python or a similar language.
  - Good understanding of programming, design, development, data modeling techniques and database fundamentals is expected as well.
  - Good understanding of modern trends in information analysis, information technology, cloud computing, object-oriented principles and agility are a plus
  - Undergraduate software engineering courses provide a good foundation

3. Outcomes expected upon the completion of the course:

- Good understanding of classical and modern lifecycle models, including agile methods
- Hands on analysis and specification skills, using methods such as features, scenarios, and user stories
- Good understanding of software as a Service, including MVC and other agile software development framework
- Practical knowledge of cloud based microservices architecture
- Understanding of architecture and design activities as well as security, privacy, testing and reliable programming
- Hands on development skills (Ruby, Rails)
- Understanding the fundamentals of Dev Ops and Code management frameworks

4. Assessment throughout the course:

- Term project execution and deliverables - content, mastery of methods discussed in class and creativity; teamwork; research and analysis skills
- Discussions - active participation and moderation of discussions; free sharing of ideas and information related to the discussion topics; systematic progress with paper reading assignments
- Quiz and assignments - content, understanding of methods discussed in class and their effective user or application to the assignment; research and analysis skills
- Class participation – open contribution to the discussions and exercises, sharing, collaboration
- Final exam – understanding of the course material and demonstrated effective application of the acquired knowledge and skills to solving practical problems

5. Required & Recommended texts:

- ***Lecture Notes***
  Lecture notes are the basic course material for this class. The notes are made available on Canvas every week.

- ***Text Book***
  - [**ESP**] Engineering Software Products, An introduction to modern software engineering, Ian Sommerville, Pearson, Hoboken, 1st Edition, 2020. https://www.amazon.com/Engineering-Software-Products-Ian-Sommerville/dp/013521064X
  - [**ESS**] Engineering Software as a Service: An Agile Approach using Cloud Computing, Second Edition, 2.0a5, Armando Fox and David Patterson https://www.amazon.com/dp/1735233803

  The following books are highly recommended. Though you will not be tested on the material here, but several lectures, examples, and discussions may refer to the material here.
  - [**FIT**] Fix IT, See and solve the problems of digital healthcare (Links to an external site.), Harold Thimbleby, Oxford University Press, 2021
  - [**TES**] The Essence of Software, Why concepts matter for design, Daniel Jackson, Princeton University Press 2021, https://www.amazon.com/Essence-Software-Concepts-Matter-Design/dp/0691225389

- ***Articles and Discussion Supporting Materials***
  For the list of readings check the Course Outline available on Canvas as well as on Discussions forum.

- ***Books Recommended for Extra Reading***
  - "The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2nd Edition)," Frederick P. Brooks, 1995.
  - "The Design of Design: Essays from a Computer Scientists," Frederick P. Brooks, 2010. o "Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale
  - "Practices for Scaling Lean & Agile Development: large, Multisite and Offshore Product Development with Large-Scale Scrum," Craig Larman and Bas Vodde, Pearson Education Inc, 2010.
  - Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives (2nd Edition), Rozanski, Nick, 2011.
  - "Design Patterns / Elements of Reusable Object Oriented Software," Erich Gamma, Richard helm, Ralph Johnson, and Vlissides (known as the "Gang of 4" of "GOF"), 1994.

6. Required software/hardware:

Free and open software; NJIT supported tools and hosting environments.

7. Other Web resources:

See Class Information on Canvas https://njit.instructure.com/courses/39027

Part III: Mapping Learning Outcomes to Course Assessment

Course Learning Outcome	Measure (i.e. exam, assignment, quiz or coding)
Good understanding of classical and modern lifecycle models, including agile methods	In class and online discussions; term project
Hands on analysis and specification skills, using methods such as features, scenarios, and user stories	Assignments, Quiz; term project
Good understanding of software as a Service, including MVC and other agile software development framework	In class and online discussions; assignments, term project
Practical knowledge of cloud based microservices architecture	In class and online discussions; term project
Understanding of architecture and design activities as well as security, privacy, testing and reliable programming	In class and online discussions; term project, final exam
Hands on development skills (Ruby, Rails)	In class and online discussions; term project, assignments
Understanding the fundamentals of Dev Ops and Code management frameworks	In class and online discussions; final exam

Part IV: Course Outline (Note: this course outline is preliminary and subject to change)

Week	Lecture/Activity/Discussion	Reading (preliminary) *Check Canvas for additional reading in every module.*
Week 1 Sep 4	Course logistics and introduction Course introduction – topics, objectives, Software as a Service, Agile Development, and Cloud Computing Reading: 1st reading assigned Class presentation schedule finalized	Text Book: [ESP] CH 1, [ESS] CH 1 'What is Product Line Engineering?' 'The Agile Mindset' Class presentation: Best Practices (discussion on Canvas) Every student is expected to find a paper, survey or a topic discussing one or several current best practices and to provide an outline and references on Canvas. All students are expected to comment on at least 2 postings by other students.
Week 2 Sep 11	Software as a Service: Frameworks and Languages How to Learn a New Language - Ruby Project start: 1) All groups finalized 2) Teams work together to select topic and identify project's key contributions 3) Project proposal posted on Canvas Discussion: Evolution of software development practices; review of week's one reading (paper 1 "Software Chronic Crisis") – Is there really a crisis today?	Text Book: [ESP] CH 2, [ESS] CH 2 The strengths and weaknesses of extreme programming. William Matheson Ruby Rapids (on YouTube) by Ingolf Krueger (beginner/intermediate rubykoans.com (intermediate/advanced) Try Ruby in your browser *ruby.github.io/TryRuby* Project presentation guidelines and requirements will be provided on Canvas
Week 3 Sep 18	SaAS Application Architecture: Microservices, APIs, and REST Project: 1. 1) Discussion and approval of project topics 2. 2) Market and competitors research In class discussion – projects Discussion (online): "No Silver Bullet" by Fred Brooks	Text Book: [ESP] CH 6, [ESS] CH 3 What are microservices. D.J. Speiss What is a REST API? 'Microservices'
Week 4 Sep 25	Requirements: BDD and User Stories Features and Scenarios Project meetings- Progress review Start of individual assignment 1	Text Book: [ESP] CH 3, [ESS] CH 7 How to write user stories, epics and personas Anissa Deanna 'An Introduction to Feature-driven Development'
Week 5 Oct 2	SaAS Framework : Rails as a Model-View-Controller Framework Discussion: RUP, Sacrum, agile principales Class Présentations	Text Book: [ESS] CH 4 Ruby on Rails API

	Assignment 1 due date	
Week 6 Oct 9	Testing: Test-Driven Development Class Presentations Project meetings Progress review	Text Book: [ESP] CH 9, [ESS] CH 8 A beginner's guide to testing. Philip Johnson 'How to Perform Software Product Testing' 'The Art of Agile Development: Test-Driven Development'
Week 7 Oct 16	Software Maintenance: Enhancing Legacy Software Using Refactoring and Agile Methods Class Présentations Discussion: (TBD)	Text Book: [ESS] CH 9
Week 8 Oct 23	Architecture and design basics; Design Patterns for SaaS Apps. Class Présentations	Text Book: [ESP] CH 5, [ESS] CH 4 Cloud Computing Architecture Paul Naumann 'SaaS Vs. PaaS Vs IaaS – An Ultimate Guide on When to Use What'
Week 9 Oct 30	Course Project présentations ! Class Présentations	
Week 10 Nov 6	Course Project présentations! Class Présentations	
Week 11 Nov 13	SaAS Framework: Advanced Programming Abstractions Exercises: Simple OOA models explained Start of individual assignment 2	Text Book: [ESS] CH 5, [ESS] CH 11
Week 12 Nov 20	Security & Privacy Discussion: (TBD) Assignment 2 due date	Text Book: [ESP] CH 7 OAuth 2 Introduction 'The Basics of Web Application Security'
Week 13 Dec 4	Reliable programming and Testing: Online discussion: SOA, cloud computing, DevOps (see Moodle) Discussion: Project retrospectives, team work, best current practices	Text Book: [ESP] CH 8 10 Tips for Clean Code Code Refactoring Derek Banas 'How to handle errors and exceptions in large-scale software projects'
Week 14 Dec 11	Dev Ops and Code Management Course summary Discussion☹Tentative): CMMI – is it still applicable? Course wrap-up and Q&A, Exam Preparation	Text Book: [ESP] CH 10, [ESS] CH 12 5 minute introduction to Git Michael Fudge 'What is DevOps' 'Continuous Integration'
Week 15 Dec 18	**Final Exam**	Closed book, comprehensive, 2 hours

Part V: Assignment Weighting (How Your Final Grade is Being Calculated?)

Assessment Item	Percentage of final grade
Term Project	35%
Quiz & Other assignments	15%
Final Exam (Comprehensive, closed book)	30%
Class participation, Discussions etc.	20%

Part VI: Delivery Mechanism

The following delivery mechanisms will be utilized:

Face-to-face lectures (or delivered via WebEx recordings)
Canvas: https://njit.instructure.com/courses/19755
Online resources (other than iTunes)

1. Lectures -
   Lecture recordings and supplemental references will be posted in Canvas weekly. In general, the lectures do not follow the textbook, and should not be considered a replacement for the textbook.

2. Quizzes - (15%)
   There will be several graded quizzes during the semester. Each quiz is meant to review and test your knowledge of the material covered over several weeks. There is a time limit for each quiz administration. The questions for each quiz are typical of those that will be found on the final exam. The online quizzes account for 10% of the course final grade.

3. Semester Long Group Project - (35%)
   One of the key learning for students in this class is the feel of real-world development experience -- The group project is extremely important for your learning in this class. Groups will provide deliverables along each sprint of the SDLC. There will be five sprints. The five deliverables account for 35% of the course grade. Further details will be provided in Canvas and class.

4. Exams - (30%)
   There will be a final exam, administered online. Questions for these exams will largely be drawn from and/or similar to the quizzes and discussion questions.

Group Work:

Group work is an important part of this course. Students will be allowed to self-select into groups (teams) of 5-6 individuals by a certain date. Students not having formed a group will be formed into separate groups (teams) by the instructor. It is expected that all students will contribute equally to the work of a group. Each group submission will include a cover page affirming such.

Recognizing that the semester long group project requires a sustained commitment of all members. If a group finds that a member is not carrying his/her assigned group responsibilities, the group will be able to petition the instructor to have that individual removed. Before this petition is accepted, the group and instructor will meet to discuss. If an individual is removed from a group, he/she will have to complete project steps individually. Please take group responsibilities seriously.

Unexcused late assignment submissions may not be accepted or accepted with penalty.

Part VII: Plagiarism and Academic Integrity

The approved "University Code on Academic Integrity" is currently in effect for all courses. Should a student fail a course due to a violation of academic integrity, they will be assigned the grade of "XF" rather than the "F" and this designation will remain permanently on their transcript.

All students are encouraged to look over the University Code on Academic Integrity and understand this document. Students are expected to uphold the integrity of this institution by reporting any violation of academic integrity. The identity of the student filing the report will be kept anonymous.

NJIT will continue to educate top tier students that are academically sound and are self-disciplined to uphold expected standards of professional integrity. **Academic dishonesty will not be tolerated at this institution.**

Part VIII: Getting Help - General

The IST Helpdesk is the central hub for all information related to computing technologies at NJIT. This includes being the first point of contact for those with computing questions or problems.

There are three ways to contact the Helpdesk:
1. Call 973-596-2900. Monday - Friday 8 am - 7 pm.
2. Go to Student Mall Room 48. Monday - Friday 8 am - 7 pm
3. Log a Help Desk Service Request online - https://ist.njit.edu/support/contactus.php.

Part IX: Getting Help – Canvas, WebEx etc.

PROTOCOL FOR DISTANCE LEARNING

● All instruction shall be delivered through Canvas. All directions and expectations for students will be posted on individual course Canvas pages. Students are responsible for checking each class on Canvas each day and for being especially attentive to email during this time.
● Attendance, participation, engagement, and understanding will all be monitored through submitted work.
● Work will be self-paced and guided by deadlines as designated. Coursework may be assigned and due in "chunks" so as to allow students to work through the material at their own pace while at home.
● All coursework will be submitted to Canvas.

Web-ex participant etiquette

The following participant expectations should be the norm. Students who don't follow these guidelines can be removed from the Webex meeting if necessary.

- Be on time
- Mute your microphone if you aren't talking
- Keep your video on throughout the class. In case of bandwidth issue, the instructor will turn off his video first.
- Raise virtual hand, if you want to speak on a point. You may also be asked specifically to comment on a topic.
- There will be online questions and/or surveys during the class that will be available for a limited duration. Please respond to them as and when asked. Some of these questionnaires will count towards the class participation grades.
- Only post chat messages relevant to the lessons