

# SOFTWARE ENGINEERING

## CS-UH 2012, SPRING 2025



### **Lecture 1: Course Logistics & Introduction to SE**

Dr. Mohamad Kassab  
[m.kassab@nyu.edu](mailto:m.kassab@nyu.edu)



# LECTURE OUTLINES

- I. Course Logistics
- I. Introduction to Software Engineering

## Getting to Know Me:

**Professor Mohamad Kassab**

### Contact Info:

Email: [m.kassab@nyu.edu](mailto:m.kassab@nyu.edu)

Location: C1 - 112

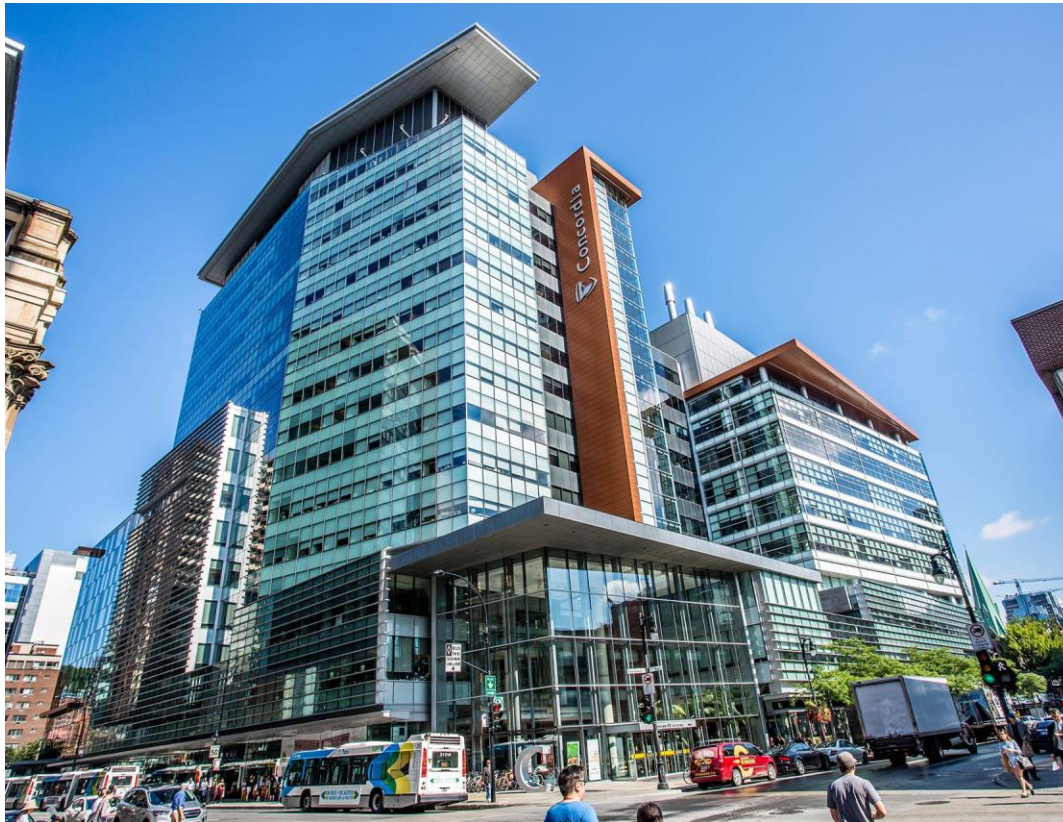
### Office Hours:

- Thursdays 11:00 AM-12:00 PM
- Or by Appointment

To make the most of our time, come prepared with specific questions you'd like to discuss during our meeting.



# EDUCATION



- **Concordia University**, Montreal, Canada
- PhD. in Computer Science, 2009
- **Concordia University**, Montreal, Canada
- Master's in computer science, 2005
- **University of Windsor**, Windsor, Canada
- Bachelor of Science: Computer Science, 2003
- **Lebanese American University**, Lebanon
- Bachelor of Engineering: Computer Engineering, 2000

# Academic Experience



Adjunct Associate Professor in CS  
(2024 – Current)

Visiting Associate Professor in CS  
(2023 – Current)



Associate Professor in SE  
(2012 – 2024)

Visiting Scholar  
(2014 – 2016)



Postdoctoral Researcher in SE  
(2010 – 2012)

Adjunct Assistant Professor  
(2010 – 2012)



Visiting Research Assistant  
(2006 – 2007)



# Industrial Experience



**Business Unit Manager  
(2018 – 2019)**

**QA Strategy Manager (2018)**



**Senior QA Engineer  
(2012)**

**Senior QA Engineer  
(2011 – 2012)**

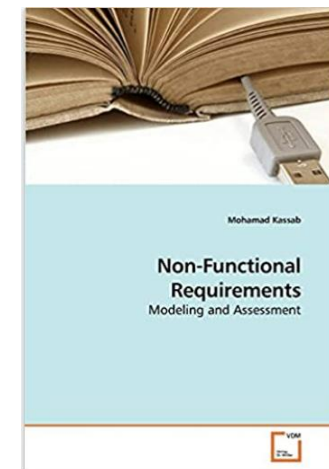
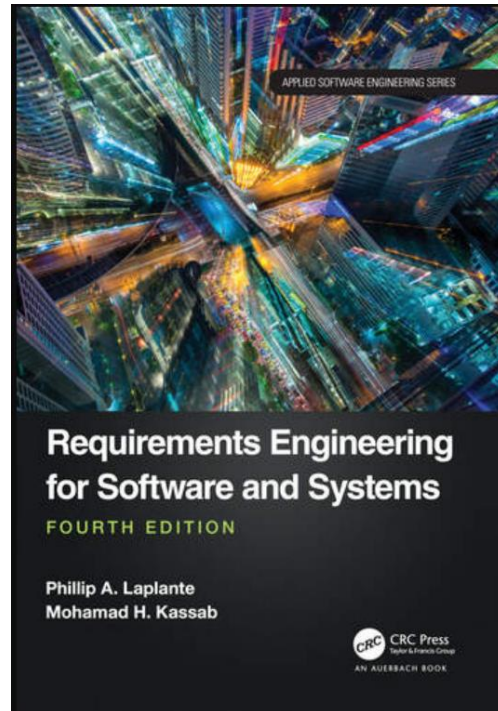
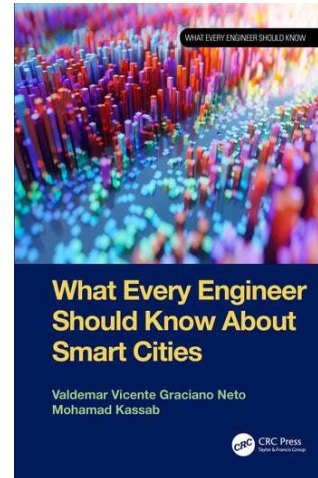
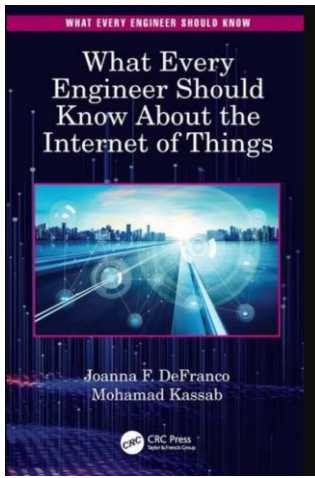


**Senior Associate  
(2010 – 2011)**

**Senior QA Specialist  
(2006 – 2010)**



**Senior Software Developer  
(2005 - 2006)**





## Course TA: FAISAL ZEESHAN

fz2335@nyu.edu

### Office Hours

- 12:00 PM – 02:00 PM
- Monday, Wednesday
- Meeting Place: A2-1111

### Booking via Calendly

- <https://calendly.com/fz2335/office-hours>

## EDUCATION

2017



**University of Bradford, UK**  
B.Sc. (Hons.) Computer Science

## EXPERIENCE

2023



**New York University Abu Dhabi, UAE**  
Computer Science Instructor

2022



**Lahore University of Management Sciences, Pakistan**  
Research Associate

2020



**Namal University, Pakistan**  
Lab Instructor

2017



**Namal Knowledge City, Pakistan**  
Senior Software Developer

## RESPONSIBILITIES

- Evaluating your course assignments and projects.
- Conducting weekly office hours to assist you with any questions.





# WHAT IS SOFTWARE ENGINEERING?



# SOFTWARE AS BLUEPRINTS

- Think of software as the blueprints that architects use to design a building.
- Just as blueprints ensure a building's stability, functionality, and aesthetics, software engineering does the same for our digital world.

# REAL-LIFE SCENARIO

- Consider a ridesharing app like Uber or Lyft.
- Have you ever wondered how these apps match you with a driver, calculate the fare, and guide them to your location?
- It's all thanks to software engineering! Every tap on your phone triggers a sequence of events orchestrated by code.



# UNMASKING SOFTWARE CATASTROPHES

## LONDON AMBULANCE SYSTEM, 1992



Further Read:  
<https://www.wired.com/2009/10/1026london-ambulance-computer-meltdown/>

13

## MARS CLIMATE ORBITER



[Watch the video in supplemental Readings.](#)

- This was one of two NASA spacecrafts in the Mars Surveyor '98 program.
- The mission failed because of software "interoperability" issue.
- The craft drifted off course during its voyage and entered a much lower orbit than planned and was destroyed by atmospheric friction.
- The metric/imperial mix-up which destroyed the craft was caused by a software error back on Earth.
- Ground software used English units, flight software used metric units – caused position calculation error – both developed by different companies.
- 350 million dollars loss and 3.5 years loss of time.

14

## THERAC 25, MEDICAL LINEAR ACCELERATOR



[Watch the video in supplemental Readings.](#)

- This was a radiation therapy machine. It was involved with at least six accidents between 1985 and 1987, in which patients were given massive overdoses of radiation, approximately 100 times the intended dose.
- Software was developed by a single person, using PDP 11 assembly language, over a period of several years. The software "evolved" from the Therac-6 software, which was started in 1972.
- The "program structure and certain subroutines were carried over to the Therac-25 around 1976."
- It appears that unit and software testing was minimal, with most effort directed at the integrated system test.

15

## TARGET CREDIT CARDS EXPOSED



- Target's systems were compromised by a stolen password
- An HVAC company had access to their networks in order to monitor the systems
- This company was actually a Pittsburgh company ...
- The perpetrators were then able to legitimately gain access to the systems
- They installed malware on the Point of Sale (POS) systems
- The malware was able to steal 40 million credit cards in 2 ½ weeks

16

## BOEING 737 GROUNDING



Further read:  
[https://en.wikipedia.org/wiki/Boeing\\_737\\_MAX\\_8\\_groundings](https://en.wikipedia.org/wiki/Boeing_737_MAX_8_groundings)

[Watch the video in supplemental Readings.](#)

17



# LONDON AMBULANCE SYSTEM, 1992



Further Read:  
<https://www.wired.com/2009/10/1026london-ambulance-computer-meltdown/>

# MARS CLIMATE ORBITER

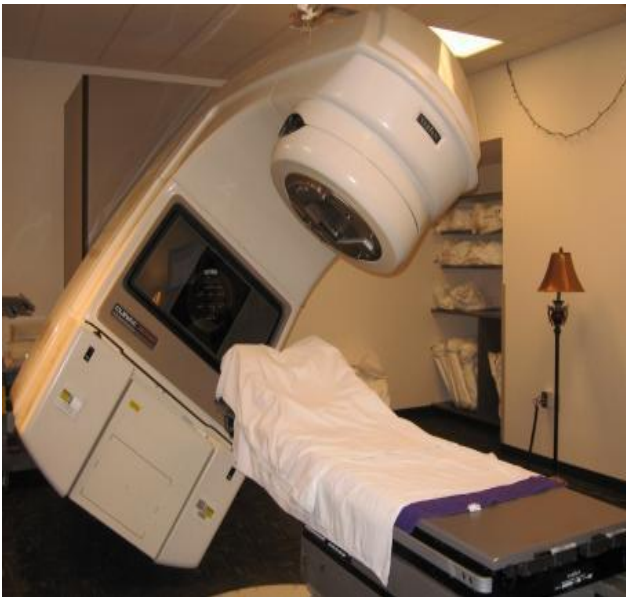


[Watch the video in supplemental Readings.](#)

- This was one of two NASA spacecrafts in the Mars Surveyor '98 program.
- The mission failed because of software “*interoperability*” issue.
- The craft drifted off course during its voyage and entered a much lower orbit than planned and was destroyed by atmospheric friction.
- The metric/imperial mix-up which destroyed the craft was caused by a software error back on Earth.
- Ground software used English units, flight software used metric units – caused position calculation error – both developed by different companies.
- 350 million dollars loss and 3.5 years loss of time.



# THERAC 25, MEDICAL LINEAR ACCELERATOR



[Watch the video in supplemental Readings.](#)

- This was a radiation therapy machine. It was involved with at least six accidents between 1985 and 1987, in which patients were given massive overdoses of radiation, approximately 100 times the intended dose.
- Software was developed by a single person, using PDP 11 assembly language, over a period of several years. The software "evolved" from the Therac-6 software, which was started in 1972.
- The "program structure and certain subroutines were carried over to the Therac-25 around 1976."
- It appears that unit and software testing was minimal, with most effort directed at the integrated system test.

# TARGET CREDIT CARDS EXPOSED



- Target's systems were compromised by a stolen password
  - An HVAC company had access to their networks in order to monitor the systems
  - This company was actually a Pittsburgh company ...
- The perpetrators were then able to legitimately gain access to the systems
- They installed malware on the Point of Sale (POS) systems
- The malware was able to steal 40 million credit cards in 2 ½ weeks



# BOEING 737 GROUNDING



Further read:

[https://en.wikipedia.org/wiki/Boeing\\_737\\_MAX\\_groundings](https://en.wikipedia.org/wiki/Boeing_737_MAX_groundings)

[Watch the video in supplemental Readings.](#)

# SO, WHAT EXACTLY IS SOFTWARE ENGINEERING?

[ONE DEFINITION]

It's the systematic application of **engineering principles** to *design, develop, test, and maintain* software. It's about creating *efficient, reliable,* and *user-friendly* solutions that solve real-world problems.

# SO, WHAT EXACTLY IS SOFTWARE ENGINEERING?

[ANOTHER DEFINITION]

It is a multi-layered approach, which includes *processes*, *standards*, and *techniques*, to allow software engineers to *design, develop* and *maintain* high-quality computer software in a timely manner.

# SO, WHAT EXACTLY IS SOFTWARE ENGINEERING?

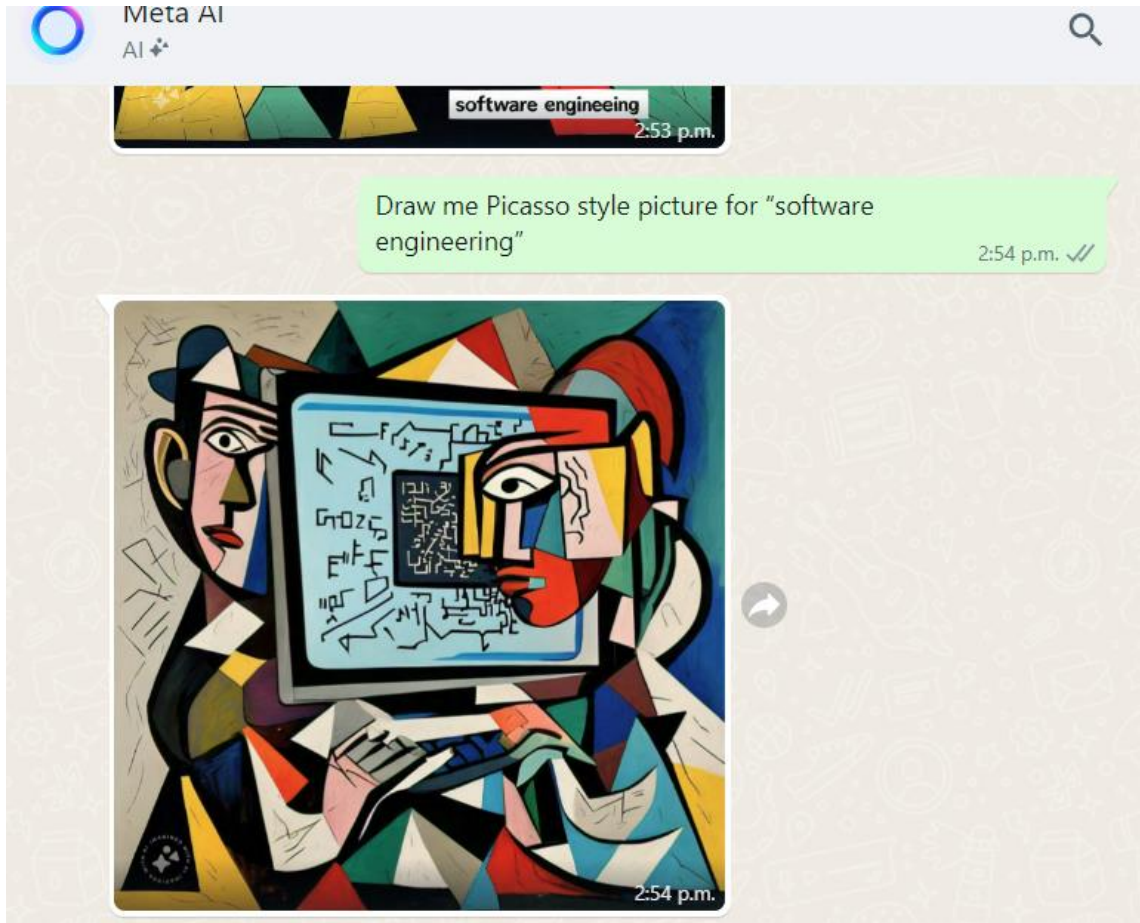
*[YET ANOTHER DEFINITION]*

It is an engineering discipline that is concerned with **all aspects of software production**

*What kind of constraints do you think you may face in the development of a software as part of a project?*

Time, security, communication and financial constraints, among others.





# META AI – IMAGINING SE: PICASSO STYLE

Explain the reason of the picture 2:54 p.m. ✓✓

The picture I generated in Picasso style for "software engineering" symbolizes the creative and innovative aspects of software development. Here's a breakdown of the elements:

- Geometric shapes: Representing code structures, algorithms, and data.
- Cubist fragments: Signifying the breaking down of complex problems into smaller, manageable parts.
- Abstract forms: Embodying the innovative and artistic nature of software engineering.
- Interconnected lines: Illustrating the connections and integrations between different components of software systems.
- Vibrant colors: Reflecting the dynamic and ever-evolving nature of technology.

This Picasso-inspired artwork captures the essence of software engineering as a fusion of art, science, and problem-solving.

2:54 p.m.

# META AI – IMAGINING SE: PICASSO STYLE

# IN CLASS EXERCISE

- Imagine you're tasked with designing an app that helps people **track their daily water intake**.
- You want to ensure it's user-friendly, accurate, and engaging.
- In pairs:
  1. Outline the key steps you would take to bring this app from concept to reality.
  2. What challenges do you foresee in each phase, and how might you address them?
- You have 10 minutes to brainstorm and create a rough plan.
- Let's discuss your ideas afterward.

# REQUIREMENTS ENGINEERING

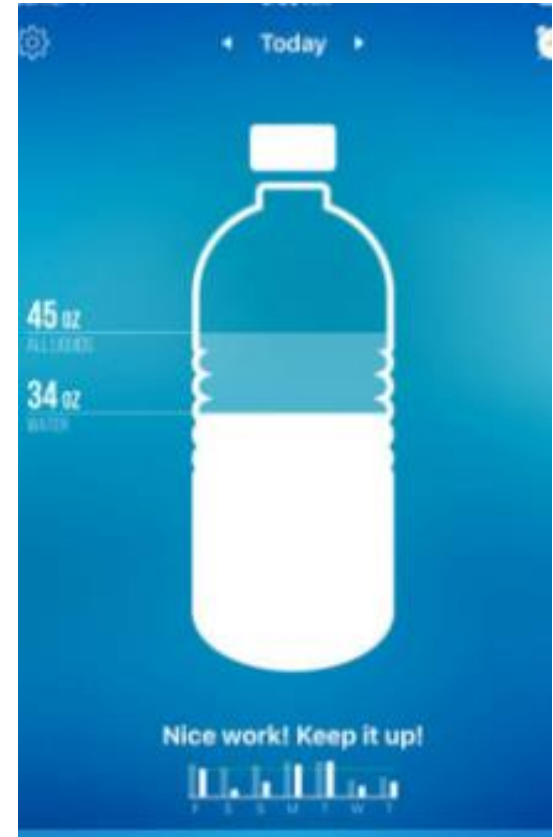
- Conduct surveys or interviews to understand user needs.
- Define features like tracking, reminders, and user-friendly interface.
- Consider platforms (iOS, Android) and compatibility.





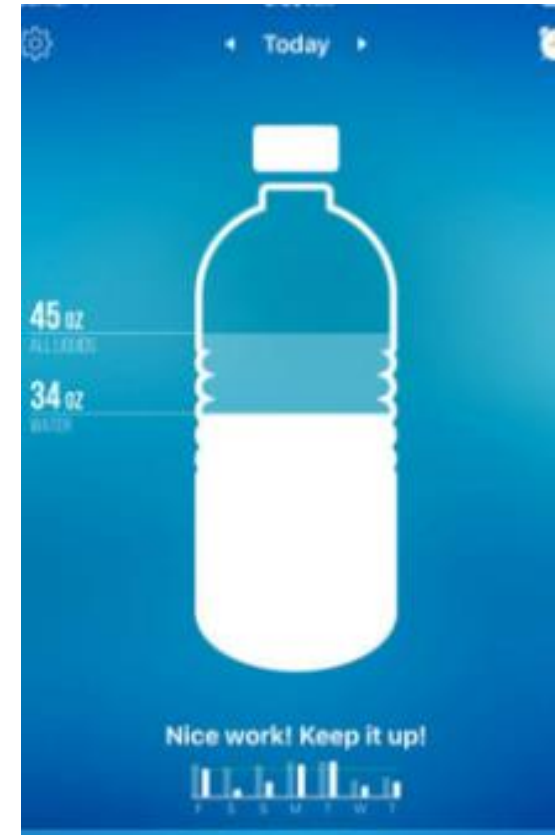
# DESIGN

- Create wireframes and mockups to visualize the app's layout.
- Design a simple and intuitive user interface.
- Plan the color scheme, fonts, and icons for a cohesive look.
- Decide on the overall architecture of the system.
- Design the classes to be implemented and assign responsibilities to classes and methods.
- Design the Database scheme.



# DEVELOPMENT

- Choose a programming language and framework (e.g., Swift for iOS).
- Implement the user interface using code.
- Develop the water intake tracking logic.



# TESTING

- Perform unit tests to ensure individual components work.
- Conduct usability testing with potential users.
- Test the app on different devices to ensure compatibility.



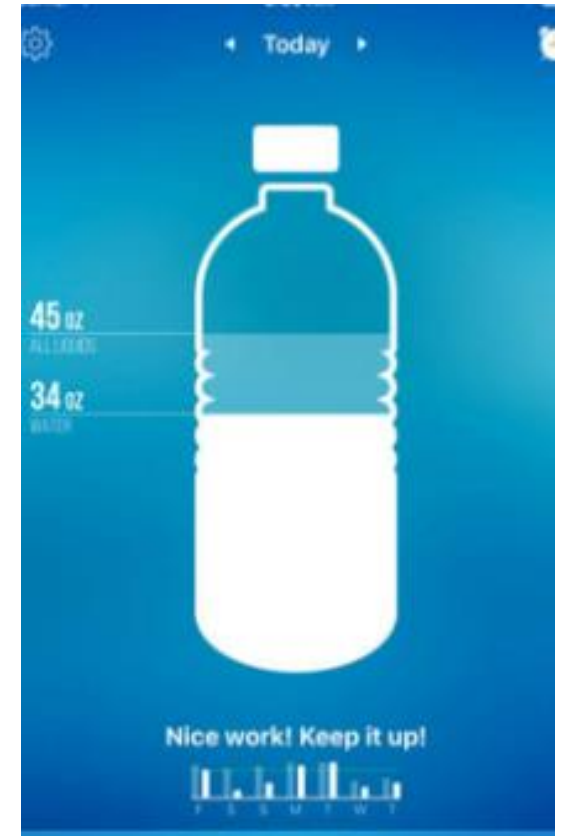
# DEPLOYMENT

- Prepare the app for release on app stores (Apple App Store, Google Play Store).
- Ensure the app meets store guidelines and requirements.
- Plan for updates and improvements based on user feedback.



# CHALLENGES AND CONSIDERATIONS

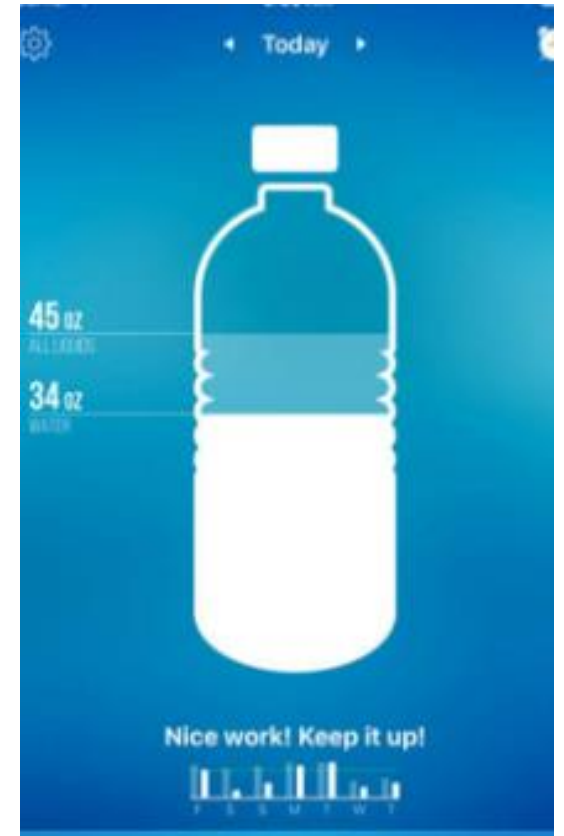
- Ensuring accurate water intake tracking.
- Addressing user privacy concerns and data security.
- Balancing features with simplicity for a user-friendly experience.
- Handling potential bugs or crashes in real-world usage.
- Supporting different devices and screen sizes.





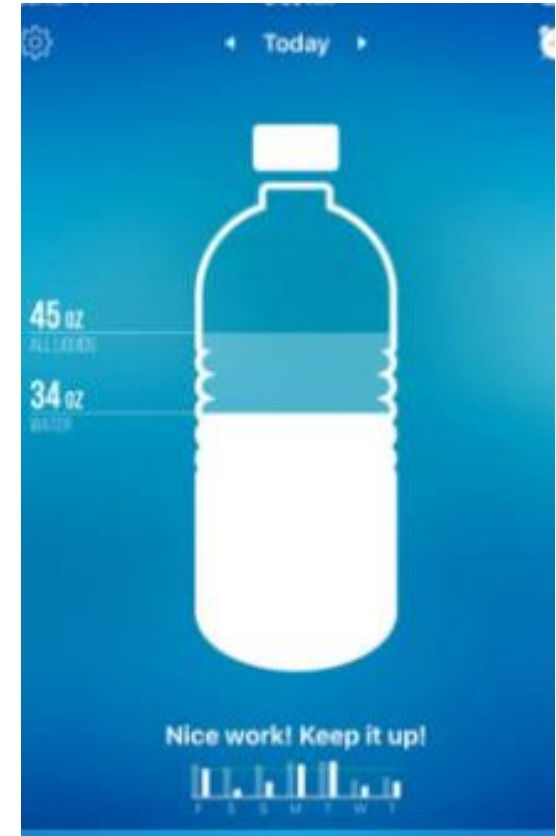
# OTHER ISSUES TO CONSIDER - COLLABORATION

- Effective communication between designers, developers, and testers.
- Regular meetings to review progress and address challenges.
- Iterative development and refining based on feedback.



# OTHER ISSUES TO CONSIDER- USER EXPERIENCE (UX)

- Prioritizing user needs to create a seamless experience.
- Designing visually appealing screens with clear interactions.
- Implementing easy navigation for users of all tech levels.



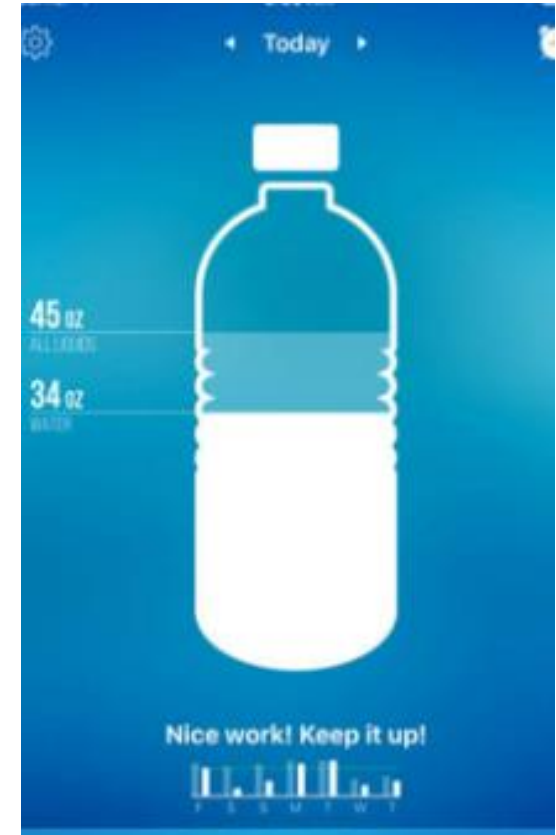
# OTHER ISSUES TO CONSIDER- PROJECT MANAGEMENT

- Creating a timeline for each phase of development.
- Identifying potential bottlenecks and allocating resources accordingly.
- Adapting the plan based on unexpected challenges and changes !



# OTHER ISSUES TO CONSIDER- SCALING AND FUTURE DEVELOPMENT

- Considering future features like integration with wearable devices.
- Planning for scalability as user base grows.
- Keeping the app updated with new OS versions and trends.



# SOUNDS EASY?

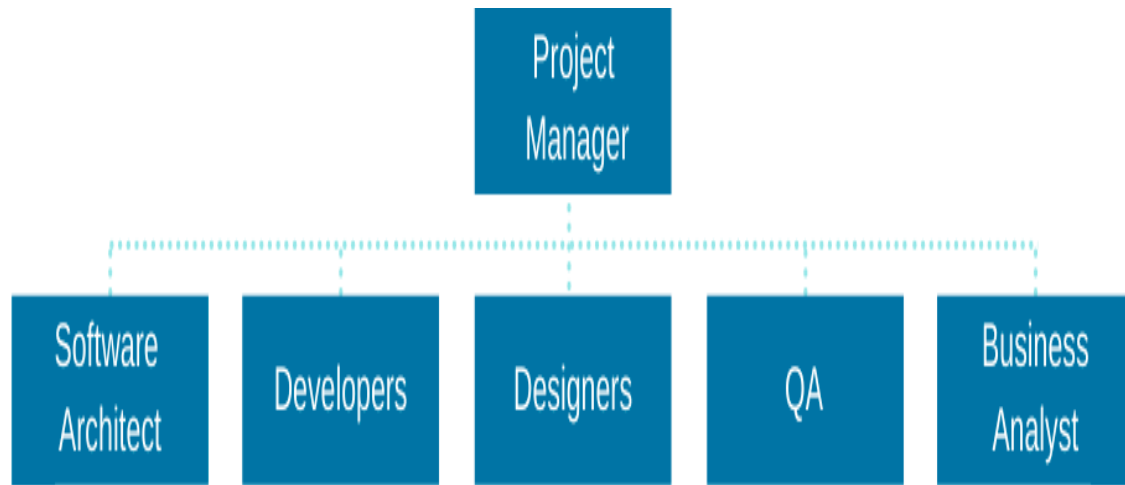
IMAGINE SOFTWARE PROJECTS ON LARGER SCALE:

- Complex global trading system that handles billions of dollars of trades every day.
- Software embedded in a jet fighter.
- Software in a robotic cardiac surgery.



<b>SOFTWARE</b>	<b>LINES OF CODE</b>
<b>Program for computer science assignment</b>	<b>100 - 2,500</b>
<b>iOS simple game app</b>	<b>5.000 - 10,000</b>
<b>Complex desktop application (e.g., Photoshop C6)</b>	<b>4M - 5M</b>
<b>Embedded systems within automobile</b>	<b>50M - 100M</b>
<b>All Google internet services</b>	<b>2B</b>

**SCALE OF SOFTWARE SYSTEMS**



# SOFTWARE ENGINEERING TEAM

- It takes a team!
- SE involves wider responsibilities than simply the application of technical skills.
- A popular example of a team composition

# COURSE DESCRIPTION

- This course is a hands-on study of the development techniques and methods of different aspects in software engineering.
- Many misinterpret software engineering for being pure system coding, which is far from reality!
- You will learn throughout this course that software engineering is much more than that.
- Numerous procedures and steps come before, during and after implementation or coding.

# COURSE TOPICS

- Please refer to the course syllabus.





# COURSE TEXTBOOKS

- **Textbook:**

- Roger S Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill Higher Education; 9<sup>th</sup> edition
- **Note:** Textbooks are available for collection at bookstore A6 Bldg., rooms 004 & 005.

- **Other Supporting Textbooks:**

- B. Bruegge and A.H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns, and Java, Pearson, 3rd edition
- I. Sommerville, Software Engineering, Pearson, 10th edition



Activity Detail	Grade Percentage	Submission Date/Week	Linked to Course Learning Outcome(s)
Class Participation	5%	Ongoing	1, 5
Assignments	20%	Ongoing	1, 2, 3, 4, 5
Group Project	25%	Ongoing	1, 2, 3, 4, 5, 6
Midterm Exam	25%	Week 8	1, 2, 3, 4
Final Exam	25%	Exam week	1, 2, 3, 4

# ASSESSMENT TASKS

- Please refer to the course syllabus/logistics on NYU Brightspace for the assignment extension, submission, grading and integrity policies

# GRADES TO LETTERS MAPPING

**NO GRADING ON A  
CURVE HERE**

Minimum Score	Grade
[95-100]	A
[90-95[	A-
[87-90[	B+
[83-87[	B
[80-83[	B-
[77-80[	C+
[73-77[	C
[70-73[	C-
[63-70[	D
[0-63[	FAIL

# ASSIGNMENTS AND GROUP PROJECT

- There will be 3 individual assignments and 1 group project with multiple deliverables.
- Late submissions will be accepted only up to 2 days late (with a penalty), afterwards zero points will be received automatically. All assignments/deliverables are due at 11:59pm on the due date. Submission will be done through NYU LMS (Brightspace).
- A 1 hour “free” buffer for technical glitches is provided. Beyond that (and up to 24 hrs), 10% will be deducted from the task grade. Beyond 24hrs and up to 48hrs, 15% will be deducted from the task grade. Students MUST follow the code of conduct and adhere to the standards of academic integrity at NYUAD. Violations will be reported to the Vice Provost for Academic Affairs



# ACADEMIC INTEGRITY

- Students MUST follow the code of conduct with the assignments and the project and adhere to the standards of academic integrity at NYUAD.
- Integrity violations, e.g., plagiarism cases, will be reported to the Vice Provost for Academic Affairs.
- In addition, it forbids capturing, reposting, sharing, or distributing in any form conversations, course materials (including presentation slides, quizzes, assignments and exams), or solutions to any of the assessment tasks.

# ACADEMIC INTEGRITY VIOLATIONS

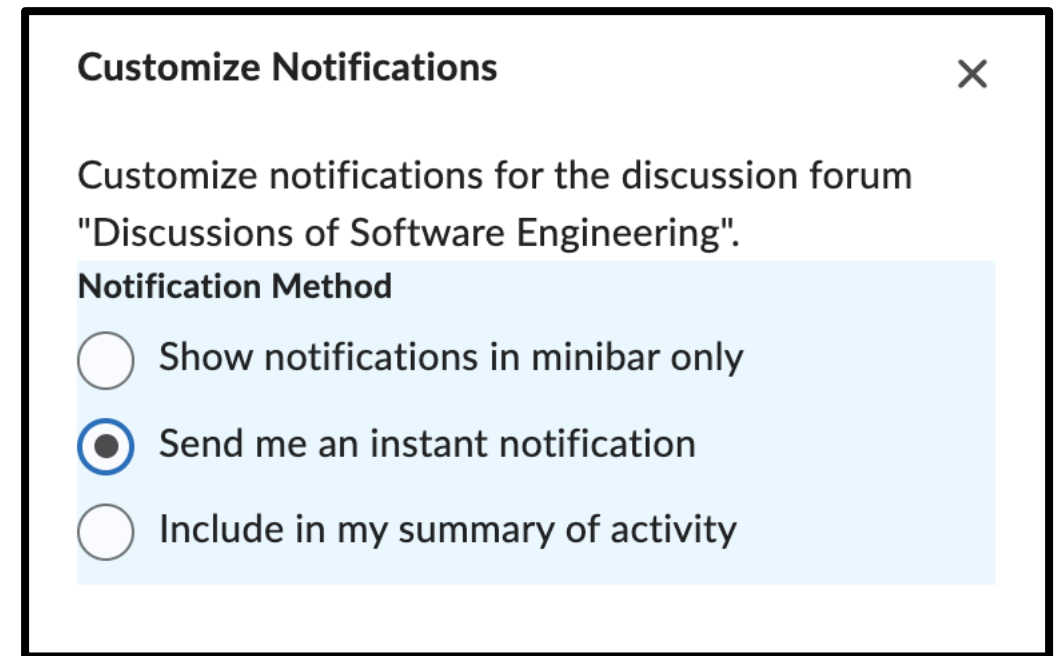
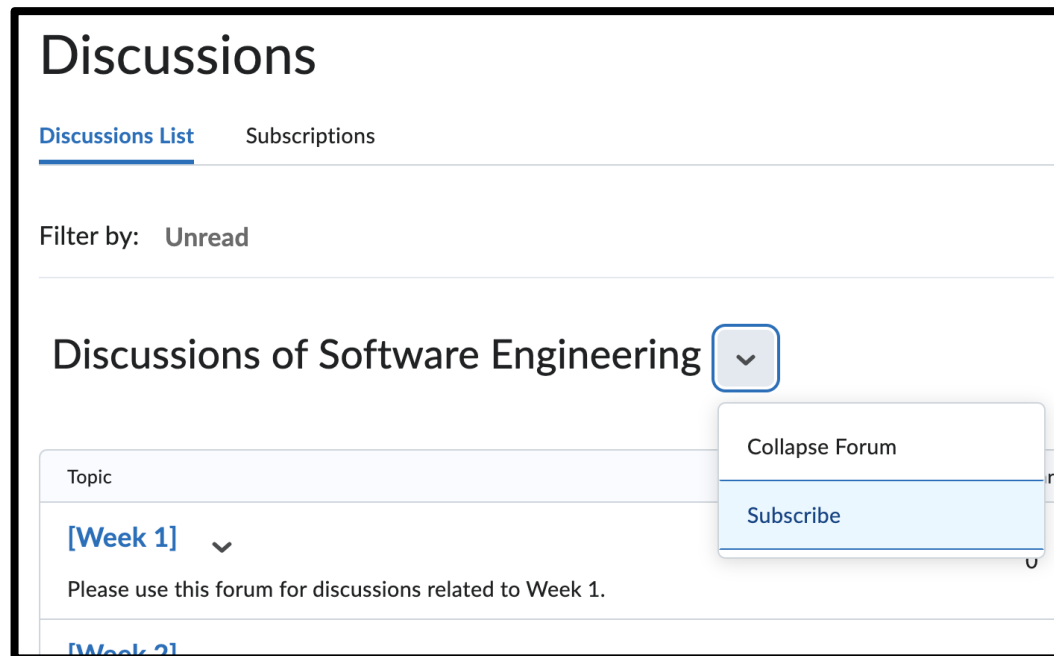
- Copying or paraphrasing someone's work is considered an integrity violation
- Permitting your own work to be copied or paraphrased, even if only in part, is considered an integrity violation
- Posting or sharing your solutions in any form whilst you are a student at NYUAD (even after you are done with the course) is considered an integrity violation
- Any submission of yours should reflect your own work. If you believe you are going to have trouble completing an assignment or with the project, please reach out to the course professor or TA at once.
- You should start working on any assignment as soon as it's released and adhere to the Assignment Submission and Extension Policies in this course

# ATTENDANCE POLICY

- Students are expected to attend all classes and be active participants in class discussions, exercises, and activities. It is essential to engage colleagues with respect in and outside the classroom. Participation will be graded based on 6 randomly selected in-class exercises/activities.
- To get the full 5% participation, students must have participated in at least 4 of these 6 exercises/activities, as well as in the two introductory and concluding course activities.

# YOU MUST SUBSCRIBE IN THE COURSE SITE'S DISCUSSIONS

For questions and updates throughout the semester:



# EMAILS FOR NOTIFICATIONS

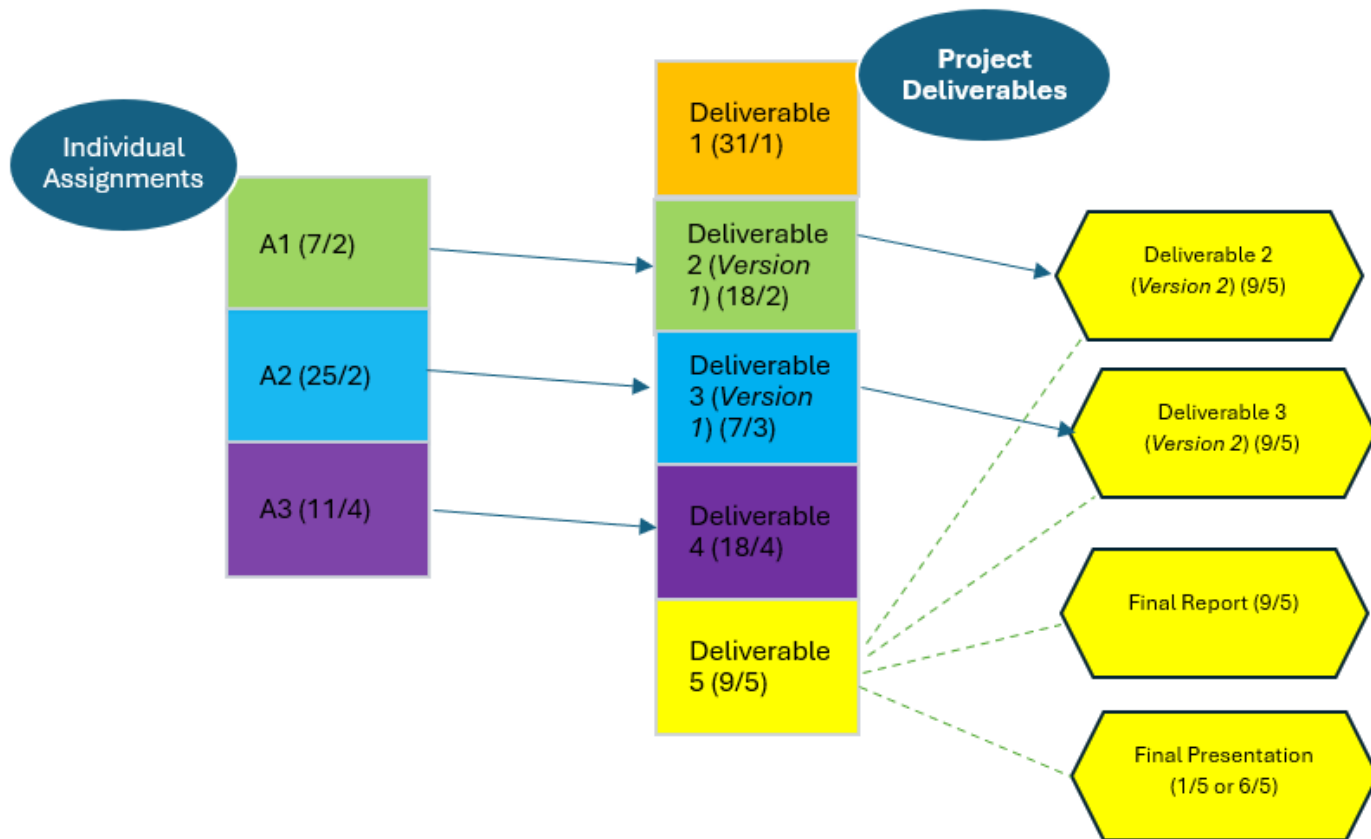
[Profile](#)  
[Notifications](#)  
[Account Settings](#)  
[Log Out](#)



## Instant Notifications

	SMS	Email
Activity Feed - new comments from others on a post	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Activity Feed - new posts created by others	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Announcements - announcement updated	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Announcements - new announcement available	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Assignments - assignment feedback released		<input checked="" type="checkbox"/>
Assignments - assignment due date or end date is 2 days away	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Assignments - assignment feedback updated		<input type="checkbox"/>
Assignments - publish all feedback completion	<input type="checkbox"/>	<input type="checkbox"/>
Content - content item created		<input checked="" type="checkbox"/>
Content - content item updated		<input checked="" type="checkbox"/>
Discussions - new post in a forum, topic, or thread that I subscribed to in instant notifications		<input checked="" type="checkbox"/>
Grades - grade item released	<input type="checkbox"/>	<input type="checkbox"/>





# INTRODUCING COURSE PROJECT

*(ALL INFO IN  
PROJECT FOLDER)*

# READINGS , ASSIGNMENTS, PROJECT TASKS



## Read:

- Chapter 1 of the textbook.
- Watch three posted videos on software failures in Supplemental Readings. (Optional).



## Assignments & Project Tasks:

- The project is open. Start forming the teams and work on Project Deliverable 1.