MET CS 673: Software Engineering
MET CS 473: Introduction to Software Engineering

Spring 2026, Section A1, On Campus
Tuesday, 18:00 – 20:45
565 Commonwealth Ave, KCB 107

Bora Eryilmaz, Ph.D.

beryilma@bu.edu
LinkedIn Profile

Course website
Blackboard website

Office Hours: Before/after class or by appointment
(Last revised on January 16, 2026)

Course Description: This course is an overview of modern techniques and tools for developing high-quality software. Topics that you will learn about include software development lifecycle management, software project planning, requirements analysis, software process frameworks, software design & architecture, code refactoring, software testing, and software integration & delivery.

The course features a semester-long *group project* where you will design and develop a software application using modern software development methodologies such as Agile and various software engineering tools, including project management & communication tools, programming frameworks, unit & system testing tools, integration & deployment tools, and version control tools.

Prerequisites: This is a *capstone course* to be taken after *at least* two programming intensive courses toward the end of a program of study. Familiarity with object-oriented design concepts and proficiency in at least one high-level programming language is *required*. Familiarity with web, desktop, or mobile application development is helpful.

Suggested Textbooks (not required):

- Roger Pressman and Bruce Maxim, *Software Engineering: A Practitioner's Approach*. 9th Edition. McGraw Hill, 2020.
- David Kung, *Software Engineering*. 2nd Edition. McGraw Hill, 2024.
- Eric J. Braude and Michael E. Bernstein. *Software Engineering: Modern Approaches*. 2nd Edition. Waveland Press, 2011.

Course Objectives: At the completion of this course, you will have better skills to:

1. Follow major software development process steps as they apply to professional software development projects.
2. Apply modern software development techniques and best practices in real-world, collaborative projects to develop high-quality software systems on time and within budget.

3. Use various software engineering tools including project management and collaboration tools, software design tools, programming tools, testing tools, and version control tools.

4. Communicate more effectively with your team members and customers during all phases of the software development process.

5. Present and discuss software process and project artifacts more clearly in both the oral and written forms.

Credit Hours: 4

Grading and Assignments: Your grade in this class will be based on participation (including your in-class and group project contributions), project reports, project presentations, the success of your group project (including code, tests, and documentation), and two in-class exams.

The grade breakdown is shown below. All percentages are approximate and the instructor reserves the right to make necessary changes. The group project success score will depend on the instructor's subjective assessment as well as the relative success of your project against all other projects in the course.

Project reports (4 documents)	25%
Presentations (mid-semester and final)	15%
Group project assessment	10%
Individual participation & contributions	10%
Midterm exam	20%
Final exam	20%

The letter vs. numerical grade conversion is shown below.

A [95-100]	A- [90-95)	
B+ [85-90)	B [80-85)	B- [77-80)
C+ [74-77)	C [70-74)	C- [65-70)
D [60-65)	F [0-60)	

- **Midterm and Final Exams**
  - There will be an in-class midterm exam and an in-class final exam.
  - The exams will be closed book and notes. You will be allowed to bring a cheat-sheet (up to four letter-size pages).
  - You are expected to work independently during the exams.
- **Group Project**
  - This course features a semester-long group project. Each team will have 4–6 members. You will be randomly assigned to a project group and you cannot change your group except for extraordinary circumstances.
  - You are expected to work as a team on your group project, each of you performing various roles over the course of the semester. However, *every member is expected to contribute an equal share to the project.*

- You will be graded on the overall success of the group project as well as your individual contributions to it. The project success will be evaluated based on the completeness, correctness, complexity, and quality of your final product (including source code, tests, design documents, and product documentation).
- Each team is required to submit a Project Proposal (as a PDF or Word file) that should be approved by the instructor before proceeding with project development.
- Each team is required to submit a number of project documents (as a PDF or Word file) describing the goals, plans, and progress of the project.
- Each team will give a short mid-semester and a final presentation of their project.
- All project artifacts should be made available on a GitHub repo (including code files, test files, design documents, product documentation, etc.). Your project documents should contain a link to your GitHub repo.

- **Individual Project Contributions**
  - Your are expected to contribute to the project as much as your other team members. *Lack of good-faith contribution to the project will negatively affect your individual contribution score.*
  - The project documents, described above, are required to contain a short section by each project team member, detailing their individual contribution to the project up to the current point in the semester.

Course Policies:

- **General**
  - Note that this is a live document and is subject to change according to the progress of the class and your feedback. Remember to check the syllabus for updates occasionally.
  - **Late return of any assignment will incur a 25% penalty per day unless pre-authorized by the instructor**.

- **Attendance and Absences**
  - Attendance and class participation is expected, but not required.
  - Your "Individual participation & contributions" score will depend on your in-class participation and your active involvement with the project work. *Simple attendance to the lectures is not sufficient to get a full score on this aspect of your grade.*
  - Students are responsible for all material covered in the lectures and by the reading assignments.

- **Academic Conduct Code**
  - Cheating and plagiarism will not be tolerated in any Metropolitan College course. They will result in no credit for the assignment or examination and may lead to disciplinary actions. Please take the time to review the Student Academic Conduct Code.
  - Use of AI tools such as ChatGPT is strictly prohibited for creating **original content** for reports, project artifacts (including code and tests), etc. You are allowed to use AI tools to **improve** any content that you have first created yourself. However, you must explicitly indicate any such use in your documents, reports, etc. *Any other, undisclosed use of AI tools in your submissions will be treated as plagiarism.*

Course Outline: The weekly coverage might change as it depends on the progress of the class. However, you must keep up with the reading assignments, projects reports, and group presentations. You can find the weekly discussion topics, lecture slides, reading assignments, and the various due dates on the Blackboard site for the course.

Week	Topics	Due
Week 1 (01/20)	• Software Engineering • Reading 1: No Silver Bullet • Reading 2: Teaching Computing Science	
Week 2 (01/27)	• Software Processes and Software Configuration Management (SCM) • Reading 1: Pro Git • Reading 2: Git & GitHub Cheat Sheet	
Week 3 (02/03)	• Software Project Management (SPM) • Reading 1: Out of the Tar Pit	• Group project proposal
Week 4 (02/10)	• Requirements Analysis	• SCMP document
Week 5 (02/17)	• No class – Substitute a Monday schedule	• SPMP document
Week 6 (02/24)	• Software Design • Reading 1: Statecharts: A Visual Formalism for Complex Systems • Reading 2: Model-View-Controller User Interface Paradigm • Reading 3: Design Patterns Card • Reading 4: Basic UML Class Diagram Notation	• SRS document
Week 7 (03/03)	• Mid-semester project presentations	• Presentation slides
Week 8 (03/10)	• No class – Spring Recess	

Week	Topics	Due
Week 9 (03/17)	• Software Architecture • Reading 1: Programming as Theory Building	
Week 10 (03/24)	• Midterm Exam • In-class, 6:00pm – 8:00pm • Location: KCB 107	• SDD document
Week 11 (03/31)	• User Experience (UX) and Interface (API) Design • Reading 1: Improving API Usability	
Week 12 (04/07)	• Software Implementation	
Week 13 (04/14)	• Software Testing – Unit Testing • Software Testing – Integration and System Testing	
Week 15 (04/21)	• Software and Digital Accessibility • Readings: Important Accessibility Websites (listed on Blackboard)	
Week 16 (04/28)	• Final project presentations	• Presentation slides
Week 17 (05/05)	• Final Exam • In-class, 6:00pm – 8:00pm • Location: KCB 107	