About Me

Education

- Bilkent University
  - Computer Science, B.S.
- Case Western Reserve University
  - Computer Science, M.S.
- Middle East Technical University
  - Information Systems, PhD

Research interests

- Software Productivity / Software Analytics
- Application Lifecycle Management
- Agile Methodologies (Scrum)
- Software Product Line Engineering

Work

- HHMI
- Microsoft
- Havelsan

@tuzuneray
@eraytuzun
About You

- https://pollev.com/erayt350
“It is not the strongest of the species that survive, nor the most intelligent, but the one most responsive to change.

Charles Darwin
My assumptions and expectations

Proficient in any programming language, but you have limited experience in analysis or design of a system

CS 201

Wants to be a Software Engineer /wants to learn

Internship experience? (poll)

For Team project
  Being a committed team member
Feedback / Class Participation

Share your thoughts
Ask questions – wave your hand forcefully – (just unmute your mic and talk) to get my attention
If there is something you do not understand -- ASK!
    There are no stupid questions!!!!
If you have a relevant comment, experience, anecdote -- SPEAK!
    Participation will make the class better!!!!
During Lecture...

- Please open up your webcams
- Try to concentrate
- Participate (Ask questions / Interact)
- No recording
Logistics

- Github (Code)
- Google docs (Documentation)
- Visual Paradigm (UML models)
- Zoom (for classes and term project group meetings)
- Polleverywhere (for polls)
- Slack for peer learning and group formation
  - Will be up soon...
Your Expectations

What are your expectations of the course?
What do you want to learn?

...

Please go to https://pollev.com/erayt350
And express your opinion...
Granularity of Software

**Trivial**: < 1 month, 1 programmer, 500 LOC
  Ex: Intro programming assignments

**Very small**: < 3 months, 1 programmer, 2000 LOC,
  Ex: Course project

**Small**: < 1 year, 3 programmers, 50K LOC,
  Ex: Mobile App

**Medium**: 3 years, 10s of programmers, 100K LOC
  Ex: Optimizing compiler

**Large**: 5 years, 100s of programmers, 1M LOC,
  Ex: MS Word, Excel, Linux, Windows

**Very large**: 10 years, 1000s of programmers, 10M LOC
  Ex: Air traffic control, Telecommunications, space shuttle
Programming != Software Engineering
Analogy with Bridge building

- Over a stream – easy, one person job
- Over river nile ... ? (The techniques do not scale)
Large-Scale Software-Engineering

Project involves a team of people – need to manage process, people and artefacts

System takes a long-time to build – need to plan

Systems are complex – need powerful tools, methods and technologies

Need to reuse code/designs/process

- > 300 Engineers
- > 10.000.000 LOC
- > 10 years!

- > $100.000.000
- > 10.000.000 LOC
Effort, Software Size, & Complexity
Course Objectives

• Learn basics of the software engineering (SE) process life cycle.

• Learn what the object-oriented (OO) approach to software development is, through OO principles and design patterns.

• Learn UML (Unified Modeling Language) that is part of most CASE (Computer Aided Software Engineering) tools and the benefits of visual modelling / diagramming.

• Practice the application of principles of object-oriented software development through the course group project.

• Develop teamwork and communication skills through the course group project.
What will you really gain from this course?

YourCV ++
- UML
- Git
- Design Patterns
- Analytical Thinking
- Requirements Analysis and Design
- Code Review
- Visual Paradigm
- Slack

Programming vs Software Engineering (Programmer -> Software Engineer)

Building software from scratch

CV writing tips
Outline

Intro to SE (Chapter 1)
Modeling w/ UML (Chapter 2)
Project Organization and Communication (Chapter 3 Sections 3.1 - 3.3)
Requirements Elicitation (Chapter 4)
Requirements Analysis (Chapter 5)
System Design (Chapters 6 & 7)
Object Design (Chapters 8 & 9)
Mapping Models to Code (Chapter 10)
Testing (Chapter 11)
New Activities (Responding to change)

- CV Development workshop
- Git Tutorial and Lab
- Design Patterns Tutorial Lab
- Teaching code review with a serious game
- Slack channel to ease up team-formation and peer learning

Please provide feedback throughout the semester related to teaching style
Contact Information

TA: Elgun Jabrayilzade, Erdem Tuna (Office: EA-527)
- Elgun (elgun@bilkent.edu.tr): Questions related to Project groups assignment, Visual Paradigm Account, Design Reports, GitLab
- Erdem (erdem.tuna@bilkent.edu.tr): Requirement Reports, Implementation & Final Report

- Direct all your project related questions to your TA first

Check website for TA’s office hours
Contact Information

Eray Tüzün

- email: eraytuzun@cs.bilkent.edu.tr
- Office hours: by arrangement
- Ask me any questions in the class / after the class
  (Prefer one-to-one QA over email)
# Course Schedule (Section 1)

## Schedule of CS 319-001

<table>
<thead>
<tr>
<th>Time</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>EE-412</strong></td>
</tr>
<tr>
<td>11:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>EE-412</strong></td>
</tr>
<tr>
<td>12:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:30</td>
<td></td>
<td></td>
<td><strong>EE-412</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:30</td>
<td></td>
<td></td>
<td><strong>EE-412</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Recommended - Textbook: Developing Software with UML, Object-Oriented Analysis and D, Bernd Oestereich, 1999, Addison-Wesley

Recommended - Textbook: Object-Oriented Analysis and Design with Applications, 2nd e, G. Booch, 1994, Benjamin/Cummings


Required - Textbook: Object-Oriented Software Engineering, Using UML, Patterns, and Java, Bernd Bruegge and Allen H. Dutoit, 2010/3rd, Pearson
Grading (Tentative)

Grading Criteria:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance/Quiz/Assignment</td>
<td>20</td>
</tr>
<tr>
<td>Project</td>
<td>40</td>
</tr>
<tr>
<td>Midterm [closed book &amp; notes] (TBD)</td>
<td>15</td>
</tr>
<tr>
<td>Final [closed book &amp; notes] (TBD)</td>
<td>25</td>
</tr>
</tbody>
</table>

Those students who fail to get a minimum of 30 (out of 75) points from the weighted average of the total grades (attendance/quiz/assignment, project, midterm exam) before the final exam will get the grade FZ. For instance, A/Q/A: 5/10, P: 20/100, M: 40/100 (0.2 * 50 + 0.4 * 20 + 0.15 * 40 = 24) fails, whereas, A/Q/A: 8/10, P: 30/100, M: 40/100 (0.2 * 80 + 0.4 * 30 + 0.15 * 40 = 34) will take the final exam.

* 20% of the course grade will be based on virtual labs (Design pattern, Code Review, Git), and possible pop-quizzes given during online lecture hours.
Plagiarism

All individual assignments must represent your own work.
No collaboration is permitted during the quizzes, the final examination, the individual labs, and the assignments. Collaboration among team members is permitted for the term project.
Plagiarism is to take and use as one’s own, or copy without acknowledgement, the works of another person. The provider of such material can be ruled equally culpable.
Term Project – 2020

* 5 teams per each game – First come first serve
Term Project 2021

• Classroom helper
  – Group formation
  – Peer review
    • Term project team member assessment
    • Reviewing the other’s group work
  – You are expected to extend the requirements after the requirement analysis
  – Best group’s software will be used next semester.

• Tools
  – Documentation: Google docs
  – Source control: Github
Term Project

Every Group Project

Does 99% of the work

Says he's going to help but he's not

Has no idea what's going on the whole time

Disappears at the very beginning and doesn't show up again til the very end

In school you have ever done
Schedule (From Last year: Will be updated)

<table>
<thead>
<tr>
<th>Date</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 15-30</td>
<td>Final Exam</td>
</tr>
<tr>
<td>May 7-15</td>
<td>Project Demos (Mithat Çoruh)</td>
</tr>
<tr>
<td>May 6</td>
<td>Iteration 2 - Project Peer Grades</td>
</tr>
<tr>
<td>May 6</td>
<td>Iteration 2 - Final Report</td>
</tr>
<tr>
<td>Apr 27</td>
<td>Iteration 2 - Project Design Report (by 23:59)</td>
</tr>
<tr>
<td>Apr 20</td>
<td>Iteration 2 - Project Analysis Report</td>
</tr>
<tr>
<td>Mar 31-Apr 3</td>
<td>Project Iteration 1 Demos</td>
</tr>
<tr>
<td>Apr 7</td>
<td>Midterm Exam</td>
</tr>
<tr>
<td>Mar 30</td>
<td>Iteration 1 - Project Final Report</td>
</tr>
<tr>
<td>Mar 21</td>
<td>Iteration 1 - Project Design Report (Please refer to the announcements for instructions, by 23:59)</td>
</tr>
<tr>
<td>Mar 7</td>
<td>Iteration 1 - Project Analysis Report (Please refer to the announcements for instructions, by Oct 25 23:59)</td>
</tr>
<tr>
<td>Feb 14</td>
<td>GitHub repository created, README.md specifies choice of project with brief description.</td>
</tr>
<tr>
<td>Feb 11</td>
<td>Project groups announced</td>
</tr>
<tr>
<td>Feb 9</td>
<td>Project and team member selection (send an email to your TA by 23:59, Barış Ardış)</td>
</tr>
</tbody>
</table>
Term Project Groups

Please form your project groups of 5 (8 groups of 5 people, 2 groups of 6 people), and email to elgun@bilkent.edu.tr on Feb 5 23:59 at latest. It is sufficient to get group email only from one group member for each group. Email should include Names, IDs, preferred choice.

The ones that do not form groups will be randomly assigned to a group.

Feb 2 Add/Drop Deadline