Welcome to CPEN 321

W1 L1:
Introduction, Goals, and Logistics

Staff

Instructor: Prof. Julia Rubin

- Lectures: Mon. 3-4:30pm, MCLD 228
  Wed. 5-6:30pm, MCLD 228
- Office hours: Mon. 4:30-5:30pm, KAIS 4053
  (or by appointment)

Staff

TAs:

- Harsha Kadiyala
- Duling Lai

- Michael Cao
- Zeyad Tamimi
- Sahar Badihi

Labs:

- Mon. 10am-12pm, MCLD 348 (Michael and Sahar)
- Tue. 3-5pm, MCLD 348 (Harsha)
- Wed. 10am-12pm, MCLD 348 (Zeyad and Sahar)

TA Office Hours:

- Mon. 12-1pm, KAIS 4095
- Wed. 2-3pm, KAIS 4095

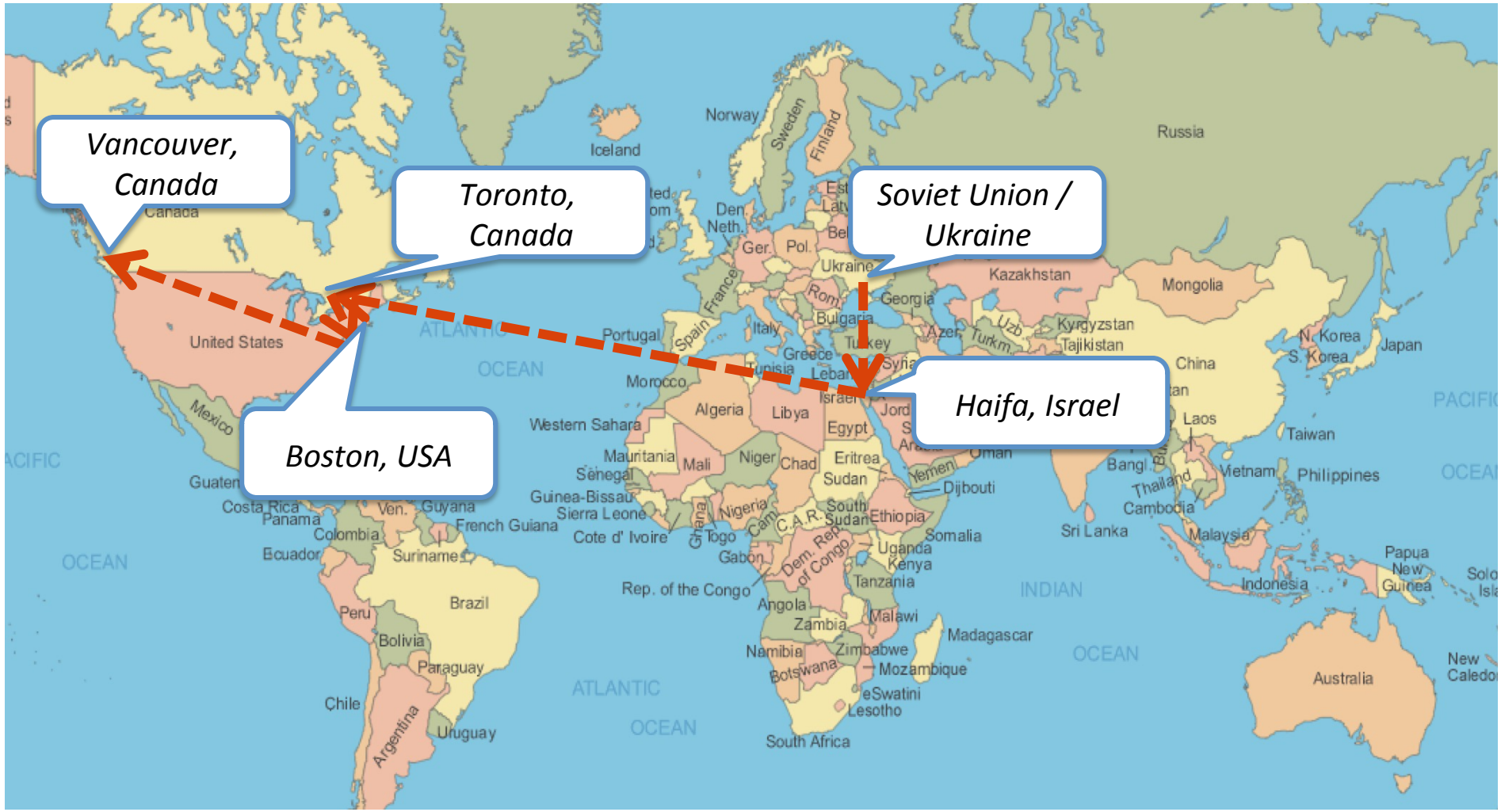
About Me

Assistant Professor of *Software Engineering* at ECE

Earlier:

- Senior Software Engineer (in a startup-company acquired by Cisco)
- Research Staff Member and Research Group Manager in IBM Research (> 8 years)
- Researcher at CSAIL, MIT
- PhD in Computer Science from University of Toronto
- MSc. and BSc. in Computer Science from the Technion

About Me


Vancouver, Canada
Toronto, Canada
Soviet Union / Ukraine
Haifa, Israel
Boston, USA
United States
Canada
Mexico
Iceland
Norway
Sweden
Finland
Russia
Kazakhstan
Mongolia
China
Japan
N. Korea
S. Korea
Taiwan
Philippines
Vietnam
Laos
Thailand
Cambodia
Malaysia
Indonesia
Papua New Guinea
Australia
New Caledonia
Sri Lanka
India
Pakistan
Afghanistan
Iran
Iraq
Saudi Arabia
Yemen
Oman
Egypt
Libya
Algeria
Morocco
Western Sahara
Mauritania
Mali
Niger
Chad
Sudan
Eritrea
Ethiopia
Somalia
Kenya
Uganda
Tanzania
Rwanda
Dem. Rep. of Congo
Rep. of the Congo
Gabon
Cameroon
Nigeria
Ghana
Togo
Cote d' Ivoire
Guinea-Bissau
Senegal
Sierra Leone
Angola
Zambia
Malawi
Zimbabwe
Mozambique
Madagascar
Namibia
Botswana
eSwatini
Lesotho
South Africa
Brazil
Peru
Bolivia
Paraguay
Chile
Argentina
Uruguay
Colombia
Ecuador
Panama
Costa Rica
Guyana
Suriname
French Guiana
Ven.
Portugal
Spain
France
Italy
Greece
Bulgaria
Romania
Ukraine
Georgia
Turkey
Syria
Lebanon
Israel
Jordan
Tunisia
Germany
Poland
Belarus
Estonia
Latvia
Denmark
Netherlands
Kyrgyzstan
Tajikistan
Uzbekistan
Turkmenistan
Azerbaijan
ATLANTIC OCEAN
PACIFIC OCEAN
INDIAN OCEAN

Research Focus

- Software engineering
- Software integrity and robustness
- Mobile and cloud-based software
- Static and dynamic code analysis

About This Course

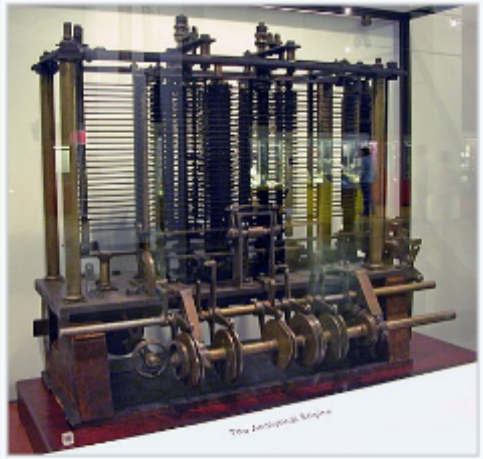
I drew inspiration and adapted part of the slides from the previous editions of CPEN 321 and CPSC 310 (University of British Columbia), as well as from CSE 403 (University of Washington), CSC C01 (University of Toronto), and CS 169 (UC Berkley)

Why Software?

Software is everywhere: in airplanes, cars, fridges, our phones

A Bit of History (aka girls can code)

An outline (algorithm) for what would have been the first piece of software was written by Ada Lovelace − English mathematician and writer in the 19th century.



Trial model of a part of the Analytical Engine, built by Babbage, as displayed at the Science Museum (London)[1]

It was written for the planned Analytical Engine.

Ada, Countess of Lovelace

Ada, Countess of Lovelace, 1840

Born	The Hon. Augusta Ada Byron 10 December 1815 London, England
Died	27 November 1852 (aged 36) Marylebone, London, England
Resting place	Church of St. Mary Magdalene, Hucknall, Nottingham, England
Known for	Mathematics Computing

A Bit of History (aka girls can code)

Margaret Hamilton – an American computer scientist and systems engineer. She was Director of the Software Engineering Division of the MIT Instrumentation Laboratory.

Hamilton in 1969, standing next to the navigation software that she and her MIT team produced for the Apollo project.

Her team worked on priority-based async. scheduling, prevented last-minute abort of the first moon landing

Margaret Hamilton

Hamilton in 1995

Born	Margaret Heafield August 17, 1936 (age 81) Paoli, Indiana, U.S.
Education	Earlham College University of Michigan
Occupation	CEO of Hamilton Technologies, Inc. Computer scientist

A Bit of History

- Hamilton coined the term “software engineering” during the Apollo space mission days:
  - During this time at MIT, she wanted to give their software “legitimacy”, just like with other engineering disciplines.
  - As a result, she made up the term “software engineering” to distinguish it from other kinds of engineering.

Margaret Hamilton @ ICSE 2018

Graduated in 1958 with a mathematics major and philosophy minor. A CEO of Hamilton Technologies. Received the NASA Exceptional Space Act Award (2003), and the Presidential Medal of Freedom awarded by Barack Obama (2016).

https://www.youtube.com/watch?v=ZbVOF0Uk5lU

Software Engineering = Programming ?

Software Engineering ≠ Programming

What is Software Engineering?

- Software engineering is about
  - ***Creating*** *and* ***maintaining*** *software systems by applying technologies and practices from computer science, project management, and other fields*.
  - ***People*** working in ***teams*** under constraints to create value for their **customers**
  - ***Processes*** rather than just the final product

Main Aspects of Software Engineering

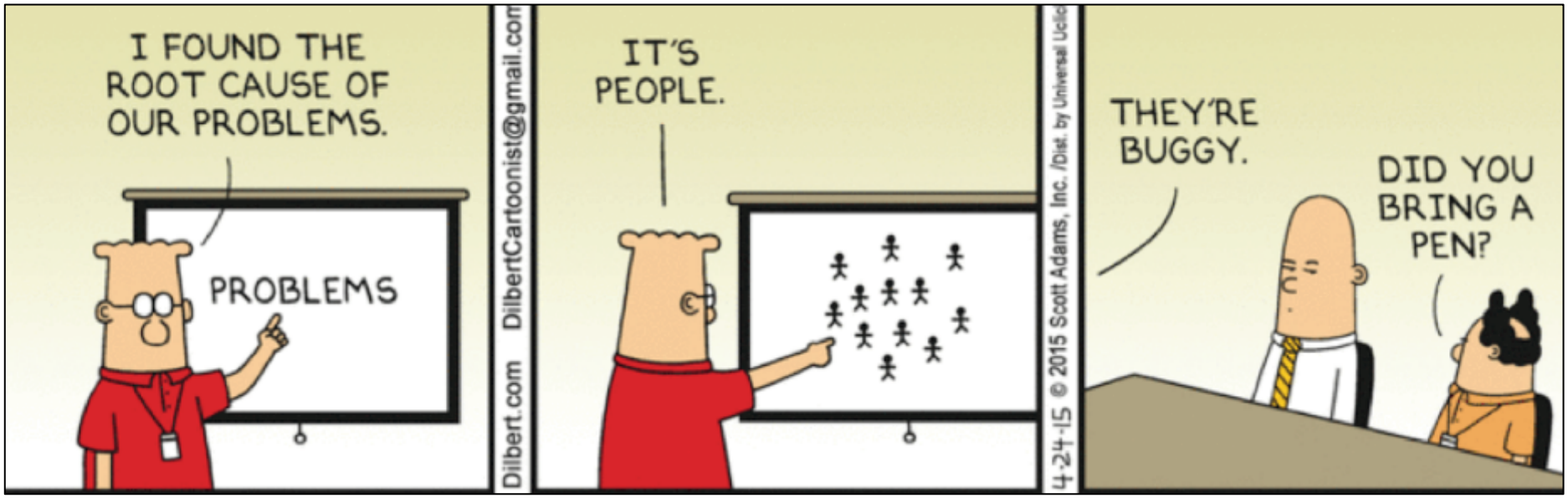
1. *Processes*, *methods*, and *techniques* necessary to turn a concept into a robust deliverable that can evolve over time.
2. Working with limited *time* and *resources*.
3. Satisfying a *customer*.
4. Managing *risk*.
5. *Teamwork* and *communication*.

Software Engineering Integrates Many Disciplines

1. Computing (programming languages, algorithms, tools)
2. Business/management (project management, scheduling)
3. Economics/marketing (selling, niche markets, monopolies)
4. Communication (managing relations with stakeholders: customers, management, developers, testers, sales)
5. Law (patents, licenses, copyrights, reverse engineering)
6. Sociology (modern trends in societies, localization, ethics)
7. Psychology (personalities, styles, usability, what is fun)
8. Art (GUI design, what is appealing to users)
9. ...

People and Software

- Customer / client: wants software built
  - often doesn't know what he/she wants
- Managers: make plans, coordinate team
  - hard to foresee all problems in advance
- Developers: design, write, and test code
  - hard to write complex code for large systems and test every combination of actions
- Marketing / sales: commit to new features
  - might require major re-engineering effort
- Users: purchase and use software product
  - misunderstand and misuse the product


I FOUND THE ROOT CAUSE OF OUR PROBLEMS.
PROBLEMS
Dilbert.com DilbertCartoonist@gmail.com
IT'S PEOPLE.
4-24-15 © 2015 Scott Adams, Inc. /Dist. by Universal Uclic
THEY'RE BUGGY.
DID YOU BRING A PEN?

Software Engineering is necessarily "*softer*" than other aspects of computing

"*softer*" ≠ easier

Software Engineering decisions require planning, justification, and management

Course Format

http://www.ece.ubc.ca/~mjulia/teaching/CPEN321-W18T1

http://piazza.com/ubc.ca/winterterm12018/cpen321

Please enroll yourself!
If you have problems with using Piazza,
please let the staff know as soon as possible.

Course Format

- *Lectures* (Mon. and Wed.) to discuss aspects of software engineering and best practices
- *Reading assignments* to reinforce concepts
- *Guest lectures* to get industrial perspective
- *Group project* for hands-on experience, and to think about technical (project scope) and social (team effort) challenges

Lecture Topics

- Software Lifecycle, Development Processes
- UML, Requirements
- Design, OO Design Patterns, REST, Microservices
- Testing and Analysis
- Code Reviews, Anti-patterns
- Teamwork, Version Control
- Continuous Integration, DevOps

Course Prerequisites

- One of EECE 210, CPEN 221, CPSC 210, EECE 309.

Grading at the High Level

- Three components:
  - Project: 55%
  - Quizzes: 40%
  - Participation: 5%

There is no final exam!

Project

- You make proposals (in teams of 3-4)
  - larger teams → larger scope
- You present your proposals.
- You find another team that likes your idea and will become your customer.
- You should like their idea and become their customer as well!

Project

- Team A proposes idea X
- Team B proposes idea Y
- A and B form a "*community*" (6-8 students)
- A implements X and acts as a client for Y
- B implements Y and acts as a client for X

Grades are given for activities in both "client" and "developer" roles

Project Scope

- A client-server software system
- *Client side*: a native mobile application (not in a web browser)
  - Android, iOS, cross-platform mobile frameworks such as React Native or Xamarin, etc.
  - Must run on a real device
- *Server side*: runs in a cloud data center
  - Microsoft Azure credit is available for interested teams
- An example of a reasonably-scoped project is available on the course web site.

Labs Next Weeks

- W2: Principles of Mobile Application Development (Android)
- W3: Principles of Backend Development

Why this kind of project?

To simulate real development settings:

- Coordination with the client
- Learning new, constantly evolving, technologies
- Collaborating with other team members
- … under tight schedule
- Technical communication skills

If you think that is hard…
(or "What is still missing?")

- Legacy (existing) code
- Collaborating with multiple other teams (each on their own schedule)
- Working across geographic locations
- Working with non-technical customers
- …

Project Management

- In Git
- Create GitHub or Bitbucket account
  - feel free to create a throw-away or anonymous account for this course
- Getting started with Git:
  - Atlassian Git Introduction
  - Shorter Atlassian Guide

Project Milestones

- **W2** (September 14): Form groups. Each group proposes an idea that they would like to implement.
  - If you do not have a group organized after this lecture, please go to Piazza and find your partners there;
  - Group members and project proposals are submitted here: https://goo.gl/forms/Yzd5NixdvG7c0avp2;
  - On September 15, we will assign you randomly to a group!
- **W3** (September 17, 19): proposed ideas are pitched in class and voted by other teams.
  - Your first chance to practice your technical communication skills), and voted by other teams.

Project Pitches

- Prepare a 3-slide, 5-minute pitch for each group
  - Vision and novelty
  - "Must have" features
  - Bonus features
  - Envisioned architecture
  - Challenges and risks

- "Sell" your startup idea to the class

Project Milestones (cont.)

- **W3** (September 21): Form communities.
  - Communities, project Git repositories, and preferred meeting times are submitted here: https://goo.gl/forms/zZrAZnO9I0hd9Blj1
  - On September 22, we will randomly assign you to a community.

Note: all students in the same community must also be in the same lab section

Project Milestones: Weekly Meetings

- Communities are required to have *regular weekly meetings*
  - to synchronize
  - to report on the work progress

Project Milestones: Weekly Meetings

- W4: Development team and the customer discuss the requirements.
- W5: M1 – Requirements (both customer and development teams).
- W6: M2 – Design (development team).
- W8: M3 – MVP (development team).
- W9: M4 - Code review (development teams).
- W10: M5 - Test plan and results (development team).
- W11: M6 - Refined specifications (both customer and development teams).
- W12: M7 - Customer acceptance test (customer team).

The exact requirements and expectations for each milestone will be provided in lectures.

Weekly Meetings: Reports

- Each meeting must be *summarized in a 1-page report*:
  - a high-level description of the progress for the week
  - plans for next week
  - major decisions and changes in the scope of the project
  - the contributions of individual team members.
- The weekly deliverable, when required, should not be part of a 1-page report, but should be attached to it.
- Bring your weekly reports and deliverables to the meeting and submit them here: https://goo.gl/forms/dYH3NYcTGS1616wI3

Meeting Schedule

- Weekly meetings will take around 40 minutes: 20 minutes for each project in the community
  - Facilitated and evaluated by the TAs.
  - A staff member will contact your community to set up a meeting during your lab section.

- In exceptional cases, meetings outside the lab hours can be arranged in coordination with the TAs.
  - However, the meetings must hold **during the same week**.

Late Deliverables

- Late deliverables will be subject to the following penalty:
  - 1st deliverable late: 20% off.
  - 2nd deliverable late: 40% off.
  - 3rd deliverable late: 80% off.
  - 4th and on deliverable late: no points given.

Project Milestones (cont.)

- **W13**: M8 - Final project presentations (in class on Monday, November 26, and during lab hours)

- **W13** (November 28, 30): top projects presentations in class; awards.

Project Milestones – Summary

- W2 (September 14): groups are formed.
- W3 (September 17, 19): proposed ideas are pitched in class
- W3 (September 21): communities are formed.
- W4: Development team and the customer discuss the requirements.
- W5: M1 – Requirements (both customer and development teams).
- W6: M2 – Design (development team).
- W8: M3 – MVP (development team).
- W9: M4 - Code review (development teams).
- W10: M5 - Test plan and results (development team).
- W11: M6 - Refined specifications (both customer and development teams).
- W12: M7 - Customer acceptance test (customer team).
- W13: M8 - Final project presentations
- W13 (November 28, 30): top projects presentations in class; awards.

Project Culture

- This is a real project
  - We expect you to build a real system
  - To be used by real people
- This is truly engineering
  - Take initiative
  - Find and solve problems yourself (as a team)
  - Programming is only part of the job …
    … but you should be able to do it!
  - Good planning, design, team work are all needed for success

Quizzes

- In-class quizzes
  - Bring your I-Clicker to every lecture!
  - Top (n-1) quizzes will be considered for the final grade
- Two mid-terms
  - *Tentatively* scheduled for October 10 and November 19
- No final exam

Grading – In Detail

- **Project – 55%**
  - Proposal: 5% (assessed based on clarity and depths of the idea)
  - M1 – M7: 5% each (35% in total)
  - M8 (final presentation): 15%
- **Quizzes – 40%**
  - In-class I-Clicker quizzes: 15%
  - Mid-term 1: 10%
  - Mid-term 2: 15%
- **Participation – 5%**
  - "Customer" behavior, i.e., providing constructive feedback to "developers"
  - Noticing and addressing concerns of your "customer" and TAs
  - Project contribution
  - Asking and answering questions in class
  - Providing good answers to questions posed by your classmates on Piazza

Absence

- Weekly meeting starting W4 are mandatory.
  - You can miss up to 2 weekly meetings.
  - For these meetings, if your work is listed in the weekly report, you will be graded with your group.
  - No points will be credited starting from the 3rd absence.
- You can miss up to 1 in-class quiz.
  - Your (top - 1) quizzes will be considered for the final grade
  - If you miss more than 1 in-class quiz or you miss a mid-term make-up will be scheduled with a request from the Engineering Student Services

More links…

- Full syllabus: http://www.ece.ubc.ca/~mjulia/teaching/CPEN321-W18T1

- Communication with course staff: http://piazza.com/ubc.ca/winterterm12018/cpen321

- Grades will be posted on Canvas.
  Check them regularly: https://canvas.ubc.ca/

Academic Integrity

- It is simple: do not cheat
- If work is to be done individually, do it by yourself
- Take part in your group's work

Learn From Others

- “Get things done early; don’t cram at the end”
- “Don’t underestimate the difficulty of learning new programming languages, frameworks, and tools”
- “Foundation of the success of our team was communication”
- “Working together (physically) was good”
- “Well-run and consistently scheduled meetings help a project a lot”
- “Thoroughly test your code and ensure that your code always passes all current tests”
- “We learned (through some pain) to do small, frequent updates. Failing to do this results in merges that can be a nightmare.”

What will I get from all that?

Fundamental skill in modern software development, including:

- Current software engineering processes and principles;
- Client-server application development paradigms;
- Ability to adapt quickly to new toolsets;
- Team work, inter- and intra-team communication;
- Management of uncertainty in customer demands;
- Efficient technical communication and presentation.

Labs Next Weeks

- W2: Principles of Mobile Application Development (Android)
- W3: Backend Development

We will poll for interest and allocate sessions later this week!

Your Next Steps

1. By September 14: form groups and submit project proposals
2. On September 17 and 19 pitch your idea in class
3. By September 21: form communities and submit your Git repositories and preferred meeting times