CMSC 12300
Computer Science with Applications 3

Tentative Syllabus

Spring 2013 Quarter

Department of Computer Science
University of Chicago

Last updated: 5/15/2013

Lecture: TuTh 4:00-5:20 (Ryerson 276)
Lab: W 3:30-4:50 (JRL A01C)

Instructor
Borja Sotomayor
borja@cs.uchicago.edu
Ryerson 151

TA

Gustav Larsson
larsson@cs.uchicago.edu
Ryerson 177

Contents of this Document

Website:
http://www.classes.cs.uchicago.edu/archive/2013/spring/12300-1/

Course description

This course is the third in a three-quarter sequence that teaches computational thinking and skills to students in the sciences, mathematics, economics, etc. The course revolves around core ideas behind the management and computation of large volumes of data ("Big Data"). Topics include (1) Statistical methods for large data analysis, (2) Parallelism and concurrency, including models of parallelism and synchronization primitives, and (3) Distributed computing, including distributed architectures and the algorithms and techniques that enable these architectures to be fault-tolerant, reliable, and scalable.

Students will continue to use R, and will also learn C++ and distributed computing tools and platforms, including Amazon AWS and Hadoop. This course includes a project where students will have to formulate hypotheses about a large dataset, develop statistical models to test those hypothesis, implement a prototype that performs an initial exploration of the data, and a final system to process the entire dataset.

CMSC 12200, or instructor's consent, is a prerequisite for taking this course.

Course organization

The development of a project requiring the analysis of a large dataset accounts for the majority of the grade in this course. Students will be provided with a list of possible projects and datasets, but are strongly encouraged to pursue a project that is of interest to them. Projects can be developed individually or in pairs. To successfully complete their project, students must understand fundamental concepts in distributed computing, parallelism, concurrency, and statistical methods suited for large volumes of data.

The course meets two times a week for lectures that provide this conceptual scaffolding. These lectures are complemented by labs and programming assignments that allow students to put these concepts into practice through the use of distributed computing tools and systems. There are no exams. The course calendar, including the contents of each lecture and project deadlines, is shown in Table 1.

Project

Throughout the quarter, students will develop a programming project that requires working with a large dataset. For the purposes of this class, we define a "large dataset" as one that (1) will not fit in the hard drive of an average desktop/laptop computer, or (2) will be processed in such a way that a single computer would take days or weeks to produce any results. The size of these datasets will typically be in the hundreds of gigabytes.

We expect students will approach the dataset in one of two ways: (1) you will formulate a series of hypotheses on the data and will write a project that analyzes the entire dataset to prove those hypotheses, or (2) you will develop an application that requires access to the entire dataset to work (e.g., an application that can interactively answer non-trivial queries or generate interesting visualizations on the dataset).

Projects can be developed individually or in pairs, although we strongly encourage you to work in pairs. Projects will be developed on GitHub, and all submissions will be done through your

GitHub repository.

The project will be broadly divided into two phases: the data exploration and prototyping phase, and the final submission phase. In the first phase, students are expected to take a subset of the dataset and do some preliminary data exploration using the techniques and tools learned in CMSC 12200. In the second phase, students will take what they learned from the data exploration to design and implement a final version that operates on the entire dataset. This second phase will involve using some of the tools and techniques covered in this course specifically.

The project will be divided into the following specific milestones. Specific guidelines and expectations for each of these milestones will be provided before each deadline.

- **Project proposal** (due Monday, April 15). A one-page proposal describing:

  1. The dataset you will be using.
  2. A description of how you intend to perform the data exploration (including what tools and techniques you expect to use) and the ideal outcome of your first prototype.
  3. A description of the final results you expect to obtain (either the hypotheses you intend to prove or disprove, or the application you expect to develop).

- **Project proposal** (in class, Thursday, April 18). You will give a 10-minute presentation on your project proposal in class. This will be an informal presentation which will not be graded; the purpose of the presentation is for everyone in the class to be aware of all the projects, and to get feedback from your peers (and possibly identify collaborations between different projects).

- **Prototype** (due Tuesday, April 30). You must push a prototype of your project to your GitHub repository. Your prototype must include specific instructions on how to obtain the data necessary to run the prototype, and on how to run the prototype itself. You should also include a brief writeup (1-2 pages) describing what you have learned about the data in the process of exploring the data and developing a prototype.

- **Prototype presentation** (in class Tuesday, May 7). You will give a 10-minute demonstration of your prototype in class. This will be an informal presentation which will not be graded.

- **Status report** (due Monday, May 20). A brief 1-2 page report specifying what progress you've made since presenting your prototype. You must also arrange to meet with the instructor during eighth week to discuss your progress.

- **Final report** (due Tuesday, June 4). You must push the final version of your project into your repository by this date, accompanied by an 8-10 page report summarizing the design and implementation of your project, and the main conclusions you have extracted from the dataset you worked with.

- **Final presentation** (in class Tuesday, June 4). You will give a 10-minute demonstration of your project in class. This presentation *will* be graded. If possible, we will try to allocate more time for the presentations at a different (non-class) time.

Table 1: CMSC 12300 Spring 2013 Schedule

Week	Date		Lecture		Lab	Due
1	Tu	Apr 02	1	Course Introduction		
	W	Apr 03			Git & GitHub	
	Th	Apr 04	2	Statistical Methods		
2	Tu	Apr 09	3	Statistical Methods		
	W	Apr 10			AWS	
	Th	Apr 11	4	Example Application I		
3	M	Apr 15				Project Proposal
	Tu	Apr 16	5	Statistical Methods		
	W	Apr 17			Data Analysis Methods	
	Th	Apr 18	6	Proposal Presentations		
4	Tu	Apr 23	7	C++		
	W	Apr 24			C++	
	Th	Apr 25	8	C++		
5	Tu	Apr 30	9	Guest Lecture: Rcpp (Dirk Eddelbuettel)		
	W	May 01			TBD	
	Th	May 02	10	C++		
6	Tu	May 07	11	Prototype presentations		Project Prototype
	W	May 08			TBD	
	Th	May 09	12	Parallelism, Concurrency		
7	Tu	May 14	13	Parallelism, Concurrency		
	W	May 15			TBD	
	Th	May 16	14	Parallelism, Concurrency		

8	M	May 20				Status report
	Tu	May 21	15	Distributed Systems		
	W	May 22			TBD	
	Th	May 23	16	Distributed Systems		
9	Tu	May 28	17	Distributed Systems		
	W	May 29			TBD	
	Th	May 30	18	C++11		
10	Tu	Jun 04	19	Project Presentations		Final report
	W	Jun 05			TBD	

Labs

Weekly labs are intended to give you a chance to practice the material we have covered in class *and* to expose you to resources that will be useful for your projects. The code you write during the labs will not be graded.

Programming Assignments

There will be three programming assignments throughout the quarter to reinforce some of the concepts and technologies covered in class (but which few, if any, students may end up using in their projects). Although there are sometimes weeks between programming assignments, we don't expect these assignments to require more than a few hours each.

Programming assignments will be announced at least one week before they are due.

Books

This course has no required textbooks.

Grading

The final grade will be divided as follows:

- *Programming Assignments*: 15% (each assignment weighted equally)
- *Project Proposal*: 15%
- *Prototype*: 20%
- *Status Report*: 15%
- *Final Report*: 25%
- *Final Presentation*: 10%

Types of grades

Students may take this course for a quality grade (a "letter" grade) or a pass/fail grade. Students will declare at any time before their final project presentation that, depending on their final grade, they want to receive a letter grade, a pass/fail grade or withdraw from the course (a *W* grade). For example, students can declare "If my final grade is a C+ or lower, I will take a *P* (Pass) instead of a letter grade and, if my grade is an *F*, I wish to take a *W*". By default, all students are assumed to be taking the course for a quality grade.

> Note: *Students taking this course to meet general education requirements must take the course for a letter grade.*

Late submissions

All student have two 24-hour extensions that may be applied to the Project Proposal or the Status Report. These extensions are all-or-nothing: you cannot use a portion of an extension and have the rest "carry over" to another extension. If extraordinary circumstances (illness, family emergency, etc.) prevent a student from meeting a deadline, the student must inform the instructor *before* the deadline.

Policy on academic honesty

The University of Chicago has a formal policy on academic honesty that you are expected to adhere to:

`http://studentmanual.uchicago.edu/academic/index.shtml#honesty`

In brief, academic dishonesty (handing in someone else's work as your own, taking existing code and not citing its origin, etc.) will *not* be tolerated in this course. Depending on the severity of the offense, you risk getting a hefty point penalty or being dismissed altogether from the course. All cases will be referred to the Dean of Students office, which may impose further penalties, including suspension and expulsion.

Even so, discussing the concepts necessary to complete assignments is certainly allowed (and encouraged). Under *no circumstances* should you show (or email) another student your code or post your solution to a web-page or social media site. If you have discussed parts of your project with anyone other than your project partner, then make sure to say so in your submission (e.g., in a README file or as a comment at the top of your source code file). If you consulted other sources, please make sure you cite these sources.

If you have any questions regarding what would or would not be considered academic dishonesty in this course, please don't hesitate to ask the instructor.

Asking questions

The preferred form of support for this course is though *Piazza* (`http://www.piazza.com/`), an on-line discussion service that can be used to ask questions and share useful information with your classmates. Students will be enrolled in Piazza at the start of the quarter.

All questions regarding assignments or material covered in class must be sent to Piazza, and not directly to the instructors or TAs, as this allows your classmates to join in the discussion and benefit from the replies to your question. If you send a message directly to the instructor or the TAs, you will get a gentle reply asking you to send your question to Piazza.

Piazza has a mechanism that allows you to ask a private question, which will be seen only by the instructors and teaching assistants. This mechanism should be used *only* for questions that apply uniquely to you.

Additionally, all course announcements will be made through Piazza. It is your responsibility to check Piazza often to see if there are any announcements. Please note that you can configure your Piazza account to send you e-mail notifications every time there is a new post on Piazza. Just

go to your Account Settings, then to Class Settings, click on "Edit Notifications" under CMSC 12300. We encourage you to select either the "Real Time" option (get a notification as soon as there are new posts) or the "Smart Digest" option (get a summary of all the posts sent over the last 1-6 hours – you can select the frequency).