Social Computing **CS5750/IS4700 Fall 2013**	**Course Syllabus** **August 25, 2013**

Location	110, West Village H
Time	Mondays, Wednesdays 2:50PM–4:30PM
Web site	`http://www.ccs.neu.edu/course/cs5750`
Forum	`http://piazza.com/northeastern/fall2013/cs5750/home`
Instructors	Prof. Alan Mislove and Prof. Christo Wilson
Contact	`{amislove,cbw}@ccs.neu.edu` (put "[cs5750]" in the subject line)
Office hours	Prof. Mislove: Mondays, 4:30PM–6:00PM, 250 West Village H Prof. Wilson: TBA
Teaching assistant	TBA
Contact	`cs5750f13-staff@ccs.neu.edu` (put "[cs5750]" in the subject line)
Office hours	TBA

DESCRIPTION

Recently, online social networking sites have exploded in popularity. Numerous sites are dedicated to finding and maintaining contacts and to locating and sharing different types of content. Online social networks represent a new kind of information network that differs significantly from existing networks like the Web. For example, in the Web, hyperlinks between content form a graph that is used to organize, navigate, and rank information. The properties of the Web graph have been studied extensively, and have lead to useful algorithms such as PageRank. In contrast, few links exist between content in online social networks and instead, the links exist between content and users, and between users themselves.

The resulting graph is used to connect and to communicate. Unlike previous networks, graphs in online social networks intermingle people and content, allow systems designers to relate the reputation of content to the reputation of users, and vice versa. It opens the door for new types of systems, new ways of solving longstanding problems, and new security attacks and vulnerabilities.

This course provides a detailed look at popular social information systems, including from online social networks (Facebook, MySpace, Orkut), blogging and microblogging platforms (LiveJournal, Blogger, Twitter), social recommendation engines (Digg, Reddit, last.fm), collaborative organization (Wikipedia), and content sharing sites (Flickr, YouTube). Coursework includes studying models (both formal and sociological) of social information systems, and the application of them both in theory and by analyzing real data from social network interactions.

The graduate version of this courses places greater emphasis on the computing infrastructure that underlies the emerging systems. Focuses on building scalable systems for managing and manipulating large amounts of data, on ensuring privacy for the users, on designing and using interfaces for third-party applications, and on leveraging the mobile nature of the access mechanisms that many users use. A course project of the students choosing will be expected.

TOPICS

The recent popularity of online social media underlies a shift in the way people connect, communicate, and share content. When designing social computing systems, one must now understand and carefully consider the structure and use of the underlying social network. This course provides a detailed look at popular social information systems. Considers questions such as: How does information spread through a social network? What mechanisms work best at encouraging collaboration? Studies models (both computational and sociological) of social information systems, and the application of them both in theory and by analyzing real data from social network interactions.

Specific topics include: History of social media, Primer on relevant sociological theory, What causes users to form links?, What do social network links represent?, How mechanisms and policies affect links, How are links used?, How does information (including opinions, fads, and political movements) spread through cyberspace?, What causes certain items to become popular (cascade)?, Why do users behave altruistically?, How do mechanisms and policies affect links?, What mechanisms work best at encouraging collaboration?, How are security and privacy concerns addressed?

LOGISTICS

The class will twice per week for 90-minute sessions. As this is a discussion-based class, you are required to attend and participate.

PREREQUISITES

The CS5750 version of this course is intended for Computer Science Master's and Ph.D. students; the IS4700 version is intended for Computer Science or Information Science B.S. students. We expect you to understand the basics of computer systems, and to have experience implementing non-trivial systems-and-networking-type projects. You should also be able to read UNIX manual pages, and be able to familiarize yourself with UNIX utilities.

This course will be partially project-centric, and *all students* will complete in projects in groups of two (or possibly three, if necessary). Thus, to succeed in this course, you must be able to work in a group. We will allow you to form your own groups, and the course staff will serve as a matching service if necessary. As you are free to choose your partner(s), we will not be sympathetic to complaints at the end of the semester about how your group-mates did not do any work.

Finally, to succeed in this course, you should have some experience with programming in UNIX development tools, as well being willing to learn how to use online APIs (e.g., Facebook, Twitter). It is also highly recommended that you become familiar with using a debugger, as this will greatly aid you in completing the projects. At a high level, you should be motivated, eager to learn, willing to work hard, and make up, on your own, any prerequisite deficiencies you may have.

GOALS

By the end of this course, we expect you to have an understanding of the computing issues faced by emerging social media platforms. We expect you to know the basic methods for network analysis, including both locally and globally-centered metrics. We also expect you to understand the techniques used to manage, query, and interpret the massive amounts of data generated by social media. Finally, we hope that you will gain an appreciation for the variety of expertise (psychology, sociology, etc) that is needed in order to successfully design, implement, use, and deploy a social media site.

TEXTBOOK

The recommended (but not required) textbook for the course is

> Easley, David and Kleinberg, Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. University Press, Cambridge, UK. 2010.

and is available freely online. Other useful texts and resources include

> Watts, Duncan J. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, 2009.
>
> Shirky, Clay. *Here Comes Everybody: The Power of Organizing Without Organizations*. Penguin Press, 2008.

Note that none of the textbooks are required; they are only there for your reference. Any recent edition of these books should suffice.

PRESENTATIONS AND SUMMARIES

This course will have a focus on reading and discussing research papers. As such, you will be required to make a approximately 20-minute presentation on one or more research papers and lead the class discussion on the paper. We expect high-quality presentations; you should expect to spend 8 to 10 hours preparing your presentation. In addition, you should come to class with a list of questions that will spark an interesting discussion.

You should also read all of the assigned papers before the class to ensure that you arrive in class prepared to take part in the discussion. To ensure that you have read the papers, you are required to submit approximately 500-word summaries of all of the papers (500 words in aggregate) by midnight before the class. This submission must be in ASCII text format (no Word, PDF, etc). In aggregate, these summaries will count for 10% of your grade.

Summaries are due at 11:59:59pm on the day before the lecture. Summaries will not be accepted late, and slip days cannot be used on summaries.

HOMEWORKS

The goal of this course is to teach the both the fundamentals of social information systems, as well as how to write programs which leverage social media. As such, there will be multiple homework programming projects throughout the semester (in addition to a course-long project, described below). You will form groups of two people to do the homeworks.[1] To collaborate effectively, you should both be involved in all of the major design decisions. You should also determine a partitioning of responsibilities so that you can both work effectively in parallel. For example, one might be responsible for generating all the test code while the other is responsible for the main code. You may switch groups between homeworks.

As the graduate version of this course will contain extra requirements, it is strongly recommended (but not required) that teams be formed of either all undergraduates or graduate students. If any of the team members are graduate students, the project will be graded as a graduate student project and will not receive any additional credit.

[1] If necessary, one group of three will be allowed.

Homeworks are due at 11:59:59pm on the specified date. You do not need to inform the instructor about the use of slip days; they will be automatically deducted.

COURSE PROJECT

You will also decide on a project of your choosing as a course-long project. The goal of the project is to conduct a miniature version of a "real" research project. You will first pick a topic, and argue in a written research proposal that this is a topic worth exploring, and that you are capable and prepared to do so. You will design and implement a solution to the problem you have chosen, and quantitatively evaluate your solution. You will then write up the results of your project in a draft final report, which, after review, you will turn into a final report. Finally, you will give a 25-minute presentation of the results of your project in class.

I emphasize that this is a "research project", and not a "programming project". Although the implementation of your solution is an essential component, it is only one aspect of the project, next to other equally important components, such as the evaluation and the presentation of your results.

You are free to choose any project topic related to this course (with instructor approval). For example, we have collected a large amount of data on online social networks (for example, a large collection of tweets from Twitter). This suggests an interesting avenue to explore, as the data has not been looked at by researchers before. Finally, if you are doing large-scale data analysis, we will try to get computing resources that are appropriate for the amount of data analysis you aim to do.

The course project is due at 11:59:59pm on December 10, 2013. No slip days may be used on the course project. The course project also has intermediate deadlines that will be announced as the course progresses.

TEAMWORK

You will form groups of two people to do the homeworks and programming projects (if necessary, one group of three will be allowed). To collaborate effectively, you should both be involved in all of the major design decisions. You may switch groups between programming projects.

Important: *You alone are responsible for finding a partner. The class forum (newsgroup) located on Piazza is a particularly good resource for this—there will be a thread there that serves exactly this purpose. Right before or right after lecture, as well as during* TA *lab hours, are also a good time to look for partners.*

We often receive complaints that somebody cannot find their partner, or that their partner continues to promise things that are never delivered. To address this concern, the policy is **you flake, you fail**. Simply put, if you disappear, or are generally not pulling your own weight at any time during the semester, you get an F in the course right then. End of story. If you don't completely flake, but are under-responsive, we reserve the right to design an appropriate (but still fair) way of redistributing points. If your partner is flaking on you, do not wait until the end of the semester to let the course staff know; let us known *immediately*.

Of course, disasters happen that may pull you away from campus. You are responsible for notifying your partner and the course staff if a major time conflict arises in your life. In the real world, you don't just disappear from your job for a week. You tell people you have to go. The same thing applies here. Likewise, if you feel you're going to need to drop this class, then do it

between projects, not in the middle of one. Dropping the course in the middle of a project may be allowed by the university, but it's extremely rude to your partner(s). Be polite.

One useful bit of advice: work together with your partner. We don't mandate pair programming, but it really works. You'll be more than twice as effective as you might if you split the work up and did it separately. Also, you'll avoid the sort of rude surprises that often arise when partners have different expectations.

FORUM

We will provide a Piazza-Based Web forum (the URL is provided at the beginning of this document) that can be used by students to ask questions and exchange wisdom while completing the homeworks and projects in this course. Please use the forum to post questions and answers that may be useful to others. Specifically, questions of the form "How do I link foo?", "Does such-and-such option in the compiler work for you?", or "What is the precise interpretation of homework question III, part b?" should be posted on the forum first. If you mail me (or other course staff) these questions, we might not be able to answer them in time.

You have to register for the Piazza forum before you can read or post items. Please do so by going to
`http://piazza.com/northeastern/fall2013/cs5750` and registering as a student.

SUBMITTING SUMMARIES, HOMEWORKS, AND PROJECTS

We will use CCIS Linux-based scripts for submitting homeworks and programming projects; instructions for using these will be included with each project. Note that submitting projects via email is *not acceptable*, and no credit will be given for these submissions, and no extensions will be granted. You can submit your projects multiple times, and we will grade the latest submission. We suggest submitting your project a few times well before the deadline to ensure there are no configuration errors.

Before you can submit any assignments, you must register for the system with your student ID. To do so, `ssh` into any CCIS Linux machine and execute

```
bash$ /course/cs5750f13/bin/register-student ID
About to register user 'USER' with student ID 'ID'.  Is this correct? [yn]
```

where `ID` is your student ID, with any leading 0s removed. For example, if your NEU student ID is 003044, you would enter

```
bash$ /course/cs5750f13/bin/register-student 3044
```

Double-check that the userid and student ID are correct; if so, type `y` and `Enter`. If not, type `n` and `Enter`. If it was successful, you will see the message

```
bash$ /course/cs5750f13/bin/register-student ID
About to register user 'USER' with student ID 'ID'.  Is this correct? [yn] y
User 'USER' is now successfully registered for CS5750 with student ID 'ID'.
bash$
```

If you see any other message (in particular, a message with "Error" in it), it has not succeeded. Email the instructor with the error message if you are not able to diagnose the problem.

To submit summaries, you will use the CS5750 turnin system described above. In particular, you should run

```
/course/cs5750f13/bin/turnin summary02 ~/path/to/file
```

if you wanted to submit the summary for lecture 02 (see the schedule for the list of assigned lecture numbers and submission keyword). When run, you should see output that looks like

```
bash$ /course/cs5750f13/bin/turnin summary02 ~/cs5750/summary02.txt
  Added file summary02.txt (28392 bytes)
Successfully submitted summary02 for user amislove (confirmation ZiwKE5).
Submitted a total of 1 files (28392 bytes) in 0 directories.
```

The script will print out every file that you are submitting, make sure that it prints out all of the files you wish to submit. Finally, make sure you see the "Successfully submitted" link at the end. You should also receive an email confirmation of your submission. If you receive any other error messages, email the instructor with the error message if you are not able to diagnose the problem.

EXAMS

There will be no exams.

GRADING

The breakdown of the grades in this course is

40%	Course project
20%	Homeworks
20%	Paper Presentation(s)
10%	Paper Summaries
10%	Participation

Each project and homework will include a breakdown and description of how it will be graded.

Any requests for grade changes or regrading must be made within 7 days of when the work was returned. To ask for a regrade, attach to your work a page that specifies (a) the problem or problems you want to be regraded, and (b) for each of these problems, why do you think the problem was misgraded.

LATE POLICY

For the homeworks, we will use flexible slip dates. Each student is given an automatic extension of 4 calendar days for the semester. You can use the extension on any homework during the semester in increments of a day.[2] For instance, you can hand in one homework 4 days late, or one homework 2 days late and two homeworks 1 day late. The slip time will be deducted from each group member's remaining slip time.[3] This should let you schedule due dates around the due dates for other courses. After you have used up your slip time, any homework handed in late will be marked off 20% per day. Homeworks more than 2 days late (beyond the use of slip days) will not be accepted. Extensions will not be granted.

ON CHEATING

[2] A "day" refers to 24 hours. Thus, a homework turned in 28 hours late will count as two days late.

[3] Note on slip days: Slip days can only be used if all group members have at least one remaining slip day. If not, the 20% off per day policy stated above will be used for all group members.

It's OK to ask someone about the concepts, algorithms, or approaches needed to do the assignments. We encourage you to do so; both giving and taking advice will help you to learn. However, what you turn in must be your own, or for the homeworks/projects, your group's own work; looking at or copying other people's code, solution sets, or from any other sources is strictly prohibited. In particular, looking at other solutions (e.g., someone else's solution to a similar project) is a direct violation. The assignments must be *entirely* the work of the students turning them in. If you have any questions about using a particular resource, ask the course staff first.

All students are subject to the Northeastern University Academic Integrity Policy, available at `http://www.northeastern.edu/osccr/academichonesty.html`. All cases of suspected plagiarism or other academic dishonesty will be referred to the Office of Student Conduct and Conflict Resolution (OSCCR).

ADVICE

As the course is difficult, it is **absolutely essential** to start early on homeworks and projects. The projects require substantial design, implementation, and testing effort, especially for students who are unfamiliar with network and systems programming. Students are encouraged to drop by the instructor's or teaching assistant's office hours (or set up a separate appointment via email) if they run into problems while completing the assignments.

Finally, computer use during class is allowed but expected to be for class purposes (e.g. note-taking, reference) only. Please avoid non-class-related computer use so that we do not have to revisit this policy.