BNFO 135: Programming For Bioinformatics

Syllabus

Instructor Info

Instructor:  Jonathan Kapleau
Office: GITC 4212
Office Phone: 973-596-2657

Course Description

The ability to use existing programs and to write small programs to access bioinformatics information or to combine and manipulate various existing bioinformatics programs has become a valuable part of the skill set of anyone working with biomolecular or genetic data. This course provides an understanding of the architecture of bioinformatics toolkits and experience in writing small bioinformatics programs using one or more of the scripting ("glue") languages frequently employed for such tasks. Python will be used for this course.

Textbooks

Think Python
Allen B. Downey
ISBN: 144933072X

Grading Scheme

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
</tbody>
</table>

9/15/2015 2:45 PM
<table>
<thead>
<tr>
<th>Final</th>
<th>30%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homeworks</td>
<td>15%</td>
</tr>
<tr>
<td>Labs</td>
<td>15%</td>
</tr>
<tr>
<td>Projects</td>
<td>10%</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10%</td>
</tr>
</tbody>
</table>

**Topics**

- Introduction to python
- Variables, expressions, & statements
- Functions
- Conditionals & recursion
- Value returning functions
- Iteration
- String processing
- Lists, dictionaries, & tuples
- Files, input & output
- Classes

**Attendance Policy**

Attendance in every lecture is mandatory. If a student is absent from lecture five times (the first day counts), the student's name will be recommended for withdrawal to the Dean of Freshman Studies. Two lates is equivalent to one absence. Make sure that you fully understand this attendance policy.

**Cheating Policy**

Cheating on a programming assignment results in zero credit for all students involved. Programming assignments may **NOT** be solved in collaboration, unless specifically stated in the assignment. Cheating on an exam will result in an "F" in the course.

You may discuss problems with each other. Where does discussion end and cheating start? You may **NOT** copy lines of code from anybody or anywhere. You may **NOT** use code in your assignments that you did not write. As a general rule: If you don't understand the code and can't explain the code, you can't use the code.

Please familiarize yourself with the NJIT Honor Code. Violations of the honor code will be dealt with seriously and reported immediately to the Dean of Students.

**Late Policy**

To receive credit, all lab assignments must have been demonstrated to the instructor on or before the due date. Assignments that are not submitted on time will not be accepted.

**Prerequisites**
CS 100 Roadmap to Computing
Course Syllabus, Fall 2015

Course Description
An introduction to programming and problem solving skills using Python, a very high level
language. Topics include basic strategies for problem solving, constructs that control the flow of
execution of a program and the use of high level data types such as lists, strings and dictionaries in
problem representation. The course also presents an overview of selected topics in computing, such
as the Internet and software engineering.

Computing is a profession that requires lifelong learning, which is pursued through activities and
using types of materials that are similar to those employed by students. In this course, each student
will track their own use of learning strategies and materials – learning how to learn efficiently in
preparation for a knowledge intensive profession.

Learning this material requires extensive hands-on practice. You should plan to spend twice as much
time studying and working problems outside of class (that is, about 6 hours a week) as you do in
class.

Course resources
Textbook: Ljubomir Perkovic, Introduction to Computing Using Python, second edition. There are
print and digital editions of the book, at different prices.

Other course materials:
• Python language V 3.4 can be gotten at https://www.python.org/downloads/ This includes the
  IDLE development environment, help files, modules and other parts of the standard
distribution. You will need to get Python and install it on your personal desktop and/or laptop
  computer. You can download Python for Windows, Mac or Linux environments. There is no
  charge for Python.
• PythonTutor, a program for stepping through and visualizing the execution of Python code at
  python tutor.com
• Optional materials: codecademy.com/tracks/python and various other online resources.

Class attendance
Class attendance is mandatory. A student who misses more than five classes will be dropped, without
credit. Getting to class late or leaving early counts as half an absence.

Weekly individual recitation
Meeting with an instructor or classroom assistant weekly is a mandatory part of the course. When
you go for recitation you should have already read the assigned material and worked on current
homework. This is an important opportunity to check whether your understanding of this material is
correct, to pose questions and get clarification. A student who misses more than three recitations will
be dropped, without credit. You may meet with the instructor or classroom assistant for your section
or for another section. All instructors and classroom assistants have posted office hours.
Homework
Homework must be submitted through Moodle and (as specified) in hard copy by the beginning of class on the due date. It will not be accepted late except for special circumstances (such as jury duty or medical problem), for which you must provide documentation.

A homework assignment will typically require you to write code that produces a specified output. No credit will be given for code that does not run. Getting a correct solution will often require that your solution be written, tested, and then rewritten multiple times until it fulfills the specification. Expect that the bulk of your time will be spent getting it right. Remember: only code that is correct is worth anything. During the write-test-debug cycle you may — and are encouraged — use the debugging facilities in the development environment, pose questions on Moodle, and discuss the problem with others.

Roadmap project
Each student will work on one or more Roadmap projects, consisting of a written and an oral presentation, either individually or with a partner. (Partners will be expected to produce two projects.)

Class participation
Presenting your homework answers and presenting your projects in class is a regular part of the course. Asking and answering questions, taking quizzes, solving programming problems — individually or in groups — is a regular part of class meetings.

Cell phones must be turned off during class. During class time you may not play games, text, email, browse the web or engage in other activities that are not part of the class.

Course communication
Moodle (http://moodle.njit.edu/) will be used to post lecture notes, to submit homework and for course discussion. You may also email instructors and classroom assistants.

Collaboration and individual responsibility
You are encouraged to study and to work on assignments together with others; collaboration is a basic learning technique. You may not take credit for the work of others. You must understand and be able to explain all work that you submit.

What You Will Learn
By the end of this course, you will be expected to know and be able to use these pieces of the computing toolkit to compute the solution of specified problems:

- Devise a problem representation (model) and a sequence of steps (algorithm) that correctly solve the problem posed
- Write a program that implements the algorithm, using
  - A core set of Python language elements (keywords, syntax, variables, modules).
  - Basic data types (integers, floats, strings, booleans, lists, dictionaries, files) and operations on them
  - Statements that perform input and output
  - Statements that control the sequence of execution of a computer program (if/else, for, while, function call/return).
Each homework assignment gives you practice on these concepts and skills, and provides feedback on your progress. You are expected to submit working solutions to every homework assignment. Each element of this course builds on previous material, and any gaps in your understanding will compromise your ability to successfully complete the course. You understand material when you are able to use it to solve problems and to explain your solutions. Each of the two midterm exams and the final exam test your mastery of the material.

**Topics to be covered**
- Starting to code – Python and IDLE
- Built-in data types
- Variables and expressions
- Conditional execution (if/else)
- Functions and methods
- Modules (turtle, math, random, url)
- Passing parameters / Returning values
- Iteration (for and while)
- Data files
- Debugging and testing
- Scope and namespaces
- User input

**Grading Formula**
- Homework 10%
- Attendance at office hours 4%
- Midterm 1 and 2, 20% each
- Final Exam 30%
- Roadmap Projects 10%
- Misc 6%

In addition to an overall course score, a minimum grade must be achieved on the final to pass the course.

**Exams**
- There are two common midterms on Monday, October 5 and Monday, November 2, 4:00-5:30 pm. Final exam date: TBA.

You must bring ID to all exams. Students with special needs are advised to make arrangements with Disability Services.

There are no makeup exams. If you miss an exam because of a documented special circumstance you may receive an imputed grade based on the other midterm.

If you believe that you deserve more credit than you have been awarded on a particular problem, you may request, within 48 hours of the exam being returned, that it be regraded. Your entire exam will be regraded, which may result in points being added or subtracted.
Exams do not require any portable electronic devices, such as a mobile phone or calculator. It is preferable that you do not bring any such device to the exam, but if you do bring one you must leave it with the proctor during the exam.

You should read the University Code on Academic Integrity (njit.edu/academics/integrity.php). It describes infractions of academic integrity and penalties for violations, including, for the most serious violations, an XF grade in the course or expulsion. All work that you represent as your own must, in fact, be your own. Work done by others must be given proper credit.

Students will be informed of any modifications of the syllabus during the semester.
CS 101: Computer Program & Problem Solving
Fall 2015

Instructor: Xiaoning Ding
Phone: (973)596-3390
E-mail: xiaoning.ding@njit.edu
Office Location: GITC 4203
Office Hours: Monday: 4:00pm ~ 6:00pm, Thursday: 4:00 pm ~ 6:00pm, additional office hours will be scheduled before exam(s) or by appointment

Teaching Assistant: Tian Tian
E-mail: tt72@njit.edu

Time and Location
Lecture:
Wednesday 1:00am ~ 2:25am (GITC 1100)

Recitation:
Section 001 --- Monday 10:00am ~ 11:25am (GITC 2400)
Section 003 --- Monday 2:30pm ~ 3:55pm (GITC 2315C)
Section 005 --- Monday 8:30am ~ 10:00am (GITC 2400)

Course Description
This course is intended for engineering freshmen not specializing in computer science who want to improve their skills on solving engineering and scientific problems. It is an introductory course that covers basic concepts of computer systems, algorithm design, programming languages, and data abstraction. The course also introduces the techniques and tools for computer programming and problem solving. The emphasis is on the logical analysis of a problem and the formulation of a computer program leading to its solution. MATLAB is used for this course.

Prerequisites
No course required.

Textbook

Course Work:
Midterm exam and final exam
Homework assignments
One project
Grading:
Your grade in the course will be determined by the following breakdown:
- Attendance – 5%
  You are expected to attend all classes and to sign the attendance sheet
- Homework assignments – 30%
  There will be 7-9 homework assignments. You are expected to finish online or submit your work through Moodle.
- Project – 10%
  One course project will be broken down into 3 steps. Each step will account for 30% of the project credits.
- Midterm – 25%
  A midterm will be scheduled in the 8th week of the semester. Close book and close notes.
- Final exam – 30%
  Final exam will be scheduled by the University. Check online for the time and location. It will be comprehensive, close-book, and close-notes.

Grading Policies:
A: 85% and above
B+: 75% and above
B: 65% and above
C+: 55% and above
C: 35% and above
D/F: remainder

Schedule
* Subject to change according to class pace

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2</td>
<td>Course overview, Chapter 1: Introduction to MATLAB</td>
</tr>
<tr>
<td>3, 4, 5</td>
<td>Chapter 2: MATLAB basics</td>
</tr>
<tr>
<td>6, 7, 8</td>
<td>Chapter 3: Branching and Program design.</td>
</tr>
<tr>
<td>9, 10</td>
<td>Chapter 4: Loops</td>
</tr>
<tr>
<td>11, 12</td>
<td>Chapter 5: User defined functions</td>
</tr>
<tr>
<td>13, 14</td>
<td>Chapter 6: Additional data types and plot types</td>
</tr>
<tr>
<td>15</td>
<td>Review</td>
</tr>
</tbody>
</table>

Course Policies

Use your NJIT email address to send emails and include “CS101” and the section number in the subject line. Your emails may be filtered out if you don’t follow.

Visit the course homepage on Moodle regularly and frequently for lecture notes, homework assignments, instructions, and latest updates.
For homework assignments and the project, all your work must be submitted electronically via the course Moodle page. Keep your work brief and to the point. You may discuss the questions in general terms with your classmates, but you must come up with your own solutions and all written work must be your own. Copying programs or written assignments (from any source, no matter your classmates or internet) is a violation of NJIT honor code.

As a general rule, no late submission will be accepted, no makeup exams will be given, and no alternate dates for exams without a legitimate reason (e.g., jury duty, medical problem). The legitimacy is determined by the office of the Dean of Students.

Collaboration of any kind is not allowed in any exams. You are not allowed to take the exam of another section. Students with special needs are advised to make arrangements with Disability Services for exam accommodations.

Turn off all mobile devices (e.g. cell phones and laptops) during class unless you are otherwise instructed.

**Honor Code**

Students must follow The NJIT Honor Code. Any violations will be brought to the immediate attention of the Dean of Students.
An introductory course in computer science and programming (using MATLAB) and its use in solving engineering and scientific problems. The emphasis is on the logical analysis of a problem and the formulation of a computer program leading to its solution. Topics include basic concepts of computer systems, algorithm design, programming languages and data abstraction. Designed for students not specializing in computer science.

1.1 Contact Information

**INSTRUCTOR:** Alex Gerbessiotis  
**OFFICE:** GITC 4213, 4th floor  
**OFFICE HOURS:** Tue 4:00-5:30pm and Thu 4:00-5:30pm. Else, by appointment Mon/Tue/Thurs  
**ASSISTANT:** TBA on course web-page  
**CLASS HOURS:** 18:00-21:05 GITC 2400  
**WEB PAGE:** http://www.cs.njit.edu/~alexg/courses/cs101/index.html

If it breaks down, use alternatively one of the following,

**WEB PAGE:** http://web.njit.edu/~alexg/courses/cs101/index.html  
**WEB PAGE:** http://cs.njit.edu/~alexg/courses/cs101/index.html

Print Handout 1 from Web-page and compare the printout to this document! They must be identical.

1.2 Course Administration

**Prerequisites**  
No course required. Knowledge of last 4 digits of NJIT ID, and NJIT UCID and password.

**Textbook**  

*Note that a newer version might be out by the start of the semester.*

**Course Work:** 2 exams (including the final); 3 MiniProjects; At most 4 unannounced Quizzes.

**Grading:**  
1000 points = Exam1(345) + Exam2(345) + QuizzesAndMP(310).

**Exams/Quizzes**  
The two exams (Exam1,Exam2) are open-textbook; you may bring a HARD-copy of the textbook but you are not allowed to borrow one during the exam. 15-minute quizzes are closed everything and worth 50 points.  
Exam1(midterm) is 90min, 345 points and Exam2(final) is 120mins, 345 points.

**MP1-3**  
Three MiniProjects; 70 points each. DUE before noon of a FRIDAY which is NOT a class day.

**Note**  
At most 310 points of Quizzes and MPs account toward the total grade. There is room to miss one (or the other).

**Due Dates**  
MiniProjects MUST BE RECEIVED BY EMAIL PER INSTRUCTIONS before NOON of the day they are due. For late submissions 30 pts deducted from grade at noon time that day, and noon the following day(s), if applicable.

**Tentative list of topics**

T1 : High-level computer organization. Introduction to computing. Bits and Bytes.  
T2 : Data representation in memory. Integers and reals  
T4 : The fundamental concepts of MATLAB. MATLAB basics  
T5 : MATLAB vector/matrix functions and operations  
T6 : MATLAB misc plotting functions  
T7 : MATLAB Branching statements  
T8 : MATLAB loops (iterative) statements. MATLAB functions  
T9 : Program design. MATLAB profiling. Recursion.  
T10: Advanced MATLAB features. Sorting and Searching.
# 2.1 Course Objectives and Outcomes

**Objective 1** Learn the fundamentals of computers, computing and programming, MATLAB and its programming environment.

**Objective 2** Learn how to use and allocate MATLAB data-types, their operations, behavior and side-effects.

**Objective 3** Learn how to trace a MATLAB program and understand its interactions with MATLAB M-files and MATLAB functions of various types and how to modify it.

**Objective 4** Learn how to use MATLAB to solve (simple) computational problems.

**Objective 5** Learn how to use MATLAB to solve more elaborate computational problems.

**Outcome 1** Be able to explain fundamental computing concepts related to processing, memory and data organization as related to engineering.

**Outcome 2** Become familiar with the syntax, functionality and capabilities of MATLAB.

**Outcome 3** Be able to understand and use MATLAB primitive data types, and effectively use built-in MATLAB functions and trace MATLAB programs.

**Outcome 4** Become familiar with matrices and arrays in MATLAB and learn how to formulate and use array operations.

**Outcome 5** Be able to provide a computer-based programming solution for simple engineering problems using a high-level language such as MATLAB and how to modify one as needed.

**Outcome 6** Be able to effectively and efficiently use MATLAB for solving more involved computational problems.

## 2.2 Tentative Course Calendar

<table>
<thead>
<tr>
<th>Week*</th>
<th>Tue to Mon</th>
<th>Exams</th>
<th>MP</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W01</td>
<td>09/01-09/07</td>
<td></td>
<td></td>
<td>Week and Semester starts on Tue (Sep 1); Note that Tue Sep 8 is a &quot;Monday&quot;</td>
</tr>
<tr>
<td>W02</td>
<td>09/08-09/14</td>
<td></td>
<td>MP1 out</td>
<td></td>
</tr>
<tr>
<td>W03</td>
<td>09/15-09/21</td>
<td></td>
<td>MP2 out</td>
<td></td>
</tr>
<tr>
<td>W04</td>
<td>09/22-09/28</td>
<td></td>
<td>MP1 due by</td>
<td>Before noon FRI OCT 02</td>
</tr>
<tr>
<td>W05</td>
<td>09/29-10/05</td>
<td></td>
<td>MP2 due by</td>
<td>Before noon FRI OCT 16</td>
</tr>
<tr>
<td>W06</td>
<td>10/06-10/12</td>
<td>EX1</td>
<td>MP3 out</td>
<td>Midterm is Ex1</td>
</tr>
<tr>
<td>W07</td>
<td>10/13-10/19</td>
<td></td>
<td></td>
<td>Mon Nov 2 : Withdrawal deadline</td>
</tr>
<tr>
<td>W08</td>
<td>10/20-10/26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W09</td>
<td>10/27-11/02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W10</td>
<td>11/03-11/09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W11</td>
<td>11/10-11/16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W12</td>
<td>11/17-11/23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W13</td>
<td>11/24-11/30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W14</td>
<td>12/01-12/07</td>
<td></td>
<td>MP3 due by</td>
<td>Before noon FRI DEC 4</td>
</tr>
<tr>
<td>W15</td>
<td>12/08-12/10**</td>
<td></td>
<td></td>
<td>Last day of classes is Thu Dec 10</td>
</tr>
<tr>
<td>FIN</td>
<td>12/15-12/21</td>
<td>EX2***</td>
<td></td>
<td>Tue Dec 15-Mon Dec 21 is Final Exam Week</td>
</tr>
</tbody>
</table>

*The Week starts on Tuesday and ends on the following Monday. ** Leftovers from Thanksgiving Recess and Labor Day. ***The final exam is prescheduled: same place, day and time as the class during exam week.

Any modifications or deviations from these dates, will be done in consultation with the attending students and will be posted on the course Web-page. It is imperative that students check the Course Web-page regularly and frequently.
Programs
Submitted code must conform to the requirements of Handout 2 and of the MiniProject(s).

Grading
If you use a pencil in an exam or quiz do not complain about grading afterwards.

Grades
Check the marks in a written work and report errors promptly. Make sure you report such problems to the grader and the instructor within two weeks from receipt, return, letter-grade posting and no later than the (first) Reading Day; for the final exam, within one week (7 calendar days) from the exam date. Talk to the grader first and then to the instructor (if different). Letter grades are decided based on a 0 to 1000 point performance. A 50% or more is C or better, 90% or more is USUALLY required for an A, but it may vary slightly depending on overall class performance.

Absenteism
The instructor reserves the right to push a student's grade down one level if he notices a student being absent from MORE than 2 classes; one way to note absenteism is absence from a quiz or not picking a graded quiz, or by handing out a sign-up list.

Collaboration
Collaboration of any kind is NOT allowed in exams or mini-projects. A student must turn in code (mini-project) that has been fully written by him/her. Any submitted code (even few lines) obtained through the Internet or otherwise, or is product of another person's/student's work, or is common with another submission in the same section/course or other, risks severe punishment, as outlined by the University; all parties of such witting or unwitting interaction receive automatically 0 in all mini-projects, not just the mini-project in question, and one lower letter grade level. The work you submit must be the result of your own effort and you must safeguard it.

Mobile Phones
Switch off (not just silence) mobile devices before class.

Email/SPAM
Always use an NJIT email address; NJIT spam filters might be unpredictable. Include cs101 in the subject line. Do not complain otherwise.

Missing class
If you miss a class and there is no Exam/Quiz it’s up to you to make up for absence.

MakeUp
There are three scheduled mini projects. Plan ahead of time and submit early; do not wait until the last day. NO EXTENSIONS are granted for any reason medical, judicial, or otherwise for the mini-projects. If you miss an exam and there is a valid documentation for your absence, such documentation must be presented to the Dean of Student Services (DOSS) within 3 working days from the day the reason for the absence is lifted and also inform us on this. The maximum accommodation will be the number of (justified) missing days to the exam date. For the quizzes no MakeUp will be given. A grade based on the Final will be extrapolated for approved cases (by the DOSS).

Final Exam
The final exam is scheduled by the Registrar; If you make travel arrangements, or make private arrangements with other instructors to have other exams rescheduled you will not be accommodated.

The NJIT Honor Code will be upheld; any violations will be brought to the immediate attention of the Dean of Student Services. Read this handout carefully!
CS 101 (13, 15, 17) Computer Programming & Problem Solving
Fall 2015

Instructors

Dr. Frank Shih, Professor
Guttenberg Information Technologies Center, Room 4205
Computer Science Department
New Jersey Institute of Technology
Newark, NJ 07102
Phone: (973) 596-5654
E-mail: shih@njit.edu
Office hours: Monday 1:00-2:25 & Thursday 1:45-2:25 in my office (GITC 4205). Also, you
can see me by appointment or send me an e-mail.

Class time:
(All sections, Lecture) Monday 2:30–3:55, GITC 1100
(Section 13, Recitation) Friday 10:00–11:25, GITC 2315C
(Section 15, Recitation) Thursday 11:30–12:55, GITC 2315C
(Section 17, Recitation) Wednesday 10:00–11:25, GITC 2400

Recitation sessions 13 & 17 will be taught by Mr. Xin Zhong (xz345@njit.edu).
Xin Zhong’s office hours:
Wednesday and Friday, 11:30 - 1:00 at Computer Vision Laboratory GITC 4324

Description of Course

This is an introductory course in computer science and programming (using MATLAB, or other
languages) and its use in solving engineering and scientific problems. The emphasis is on the
logical analysis of a problem and the formulation of a computer program leading to its solution.
Topics include basic concepts of computer systems, algorithm design, programming languages
and data abstraction. It is designed for students not specializing in computer science.

Specific Goals for the Course

- Be able to explain fundamental computing concepts related to processing, memory and
data organization as related to engineering.
- Be able to formulate succinctly and correctly the input and output relationship of
computational problems.
- Be able to provide a computer-based programming solution for technical problems using
a high-level language such as MATLAB.
- Become familiar with the syntax and functionality of MATLAB.
- Be able to effectively use MATLAB for solving more complex problems arising in
science or engineering.


Brief List of Topics to Be Covered

- High-level computer organization. Introduction to computing. Bits and Bytes.
- Data representation in memory. Integers and reals. Computational problems and input output relationship.
- The abstract data type: Matrix and Vector. Implementations using an array.
- The fundamental concepts of MATLAB. MATLAB basics. MATLAB as a calculator. Input and Output, M-files, Mat files, MATLAB diary.
- MATLAB vector/matrix functions and operations.
- MATLAB misc plotting functions.
- MATLAB branching statements.
- MATLAB iterative statements.
- MATLAB functions.
- Program design. MATLAB profiling and code optimization.
- Recursion. Tradeoffs. Searching and sorting in MATLAB.
- Advanced MATLAB features. Complex numbers, Sparse arrays, Cell arrays, structure arrays.

Course Webpage

Submit the homework solution in Microsoft Word format to http://moodle.njit.edu/ before the deadline. Absolutely, no late submission is accepted. Write the answers in your own words individually. Any plagiarism will post a “ZERO” score or cause a “FAIL” grade.

Textbook


Additional Reference


Topics

1: Starting With MATLAB
2: Creating Arrays
3: Mathematical Operations With Arrays
4: Script Files
5: Two-Dimensional Plots
6: Functions and Function Files
7: Programming in MATLAB
8: Polynomials, Curve Fitting, and Interpolation
9: Three-Dimensional Plots
10: Applications in Numerical Analysis
11: Symbolic Math

**Tentative HW/Exam Schedule**

September:
- 21: HW1 Due at 11:00PM

October:
- 5: HW2 Due at 11:00PM
- 12: Exam1
- 26: HW3 Due at 11:00PM

November:
- 9: HW4 Due at 11:00PM
- 16: Exam2
- 30: HW5 Due at 11:00PM

December:
- Final Exam (See Schedule on NJIT Website)

**Grading**

Each midterm exam will be 85 minutes long, and the final exam will be 2.5 hours long. Your grade in the course will be determined by the following breakdown:

- 5 Homework 35%
- Exam1 15%
- Exam2 15%
- Final Exam 30%
- Participation 5%

This course adopts “moodle” for homework submission. Please submit your homework solution in Microsoft Word format to http://njit2.mrooms.net/ before each deadline.

There will be a total of 100 points. The grade assign is based on the following:

- A: 90 –100 points
- B+: 80 – 89 points
- B: 70 – 79 points
- C+: 65 – 69 points
- C: 60 – 64 points
- D: 50 – 59 points
- F: 0 – 49 points

**Course Policies**

Punctuality and class attendance is mandatory. If you cannot attend some class, you must contact me beforehand. As a general rule, I do not give makeup exams, I do not allow students
to take exams on alternate dates, nor do I allow students to turn in assignments late. Of course, if someone has a legitimate reason (e.g., jury duty, serious medical problem, sports tournament, conflict with a religious holiday), I will make allowances as long as you provide proper documentation. I will not accept excuses such as having too heavy a workload or having too many exams the same week. Also, I do not give out extra-credit assignments.

For all exams, be sure to bring a photo ID. All exams will be closed book and closed notes. For exams of 90 minutes or less, no one will be permitted to leave the room once an exam has started. If a student leaves before the exam period is concluded, his/her exam will be collected and not be returned. Students with special needs are advised to make arrangements with Disability Services for exam accommodations.

If upon getting back one of your exams you think that you deserve more points on a particular problem, I will regrade the entire exam. Thus, you may get more points on the one problem, but you may lose points on other problems. Also, any questions about the grading must be asked within 48 hours of when the exam or homework was returned back.

Unless you are otherwise instructed, all portable electronic devices, such as cell phones and laptops, must be turned off during class.

Students will be informed of any modifications or deviations from the syllabus throughout the course of the semester.

**Honor Code**

Academic integrity and honesty are of paramount importance. The NJIT Honor Code will be strictly upheld, and any violations will be brought to the immediate attention of the Dean of Students. Honor Code violations include, but are not limited to,

- communicating with others during exams
- using unauthorized materials during exams
- copying/giving a computer program from/to another person.
CS 104 Computer Programming and Graphics Problems
Course Syllabus, Fall 2015
Office Hours: MW 11:30-12:30 (GITC4405)

Course Description
An introduction to programming and problem solving skills using Python, a very high level language. Topics include basic strategies for problem solving, constructs that control the flow of execution of a program and the use of high level data types such as lists and strings in problem representation. The course also presents an overview of selected topics in computing, such as the Internet and software engineering.

Course resources
Textbook: Ljubomir Perkovic, Introduction to Computing Using Python. There are color, black and white and digital editions of the book, at different prices. The content is the same.

Other course materials: Python language and integrated development environment (IDLE) python.org/download/releases/3.4.3/ ; Other materials: codecademy.com/tracks/python.

Course Policies
Attendance is mandatory.

Moodle (http://moodle.njit.edu/) will be used to post lecture notes, to submit homework and for course discussion.

Homework and projects must be submitted through Moodle by the beginning of class on the due date. They will not be accepted late except for special circumstances (such as jury duty or medical problem), for which you must provide documentation. All submitted work (including exams) must include your name, course, section and student ID.

You are encouraged to study and to work on assignments together with others, but you must always disclose collaboration and not take credit for the work of others. You must understand and be able to explain all work that you submit. Plagiarism will result in zero credit for the assignment or other penalty up to an XF grade in the course.

Presenting your homework answers and presenting your projects in class is a regular part of the course.

Cell phones must be turned off during class. During class time you may not play games, text, email, browse the web or engage in other activities that are not part of the class.

What You Will Learn
By the end of this course, you will be expected to know and be able to use these pieces of the computing toolkit to compute the solution of specified problems:

- Devise a problem representation (model) and a sequence of steps (algorithm) that correctly solve the problem posed
- Write a program that implements the algorithm, using
- A core set of Python language elements (keywords, syntax, variables, modules).
- Basic data types (integers, floats, strings, booleans, lists, files) and operations on them
- Statements that perform input and output
- Statements that control the sequence of execution of a computer program (if/else, for, while, function call/return).

**Topics to be covered**
Starting to code -- Python Programming Language  
Built-in data types  
Variables and expressions  
Conditional execution (if/else)  
Functions and methods  
Modules (turtle, math, random, url)  
Passing parameters  
Returning values  
Iteration (for and while)  
Data files  
User input

**Grading Formula**
Homework 20%  
Midterm 25%  
Final Exam 30%  
Roadmap Project 25%

There are no makeup exams. Exams do not require any portable electronic devices, such as a mobile phone or calculator.

You should read the University Code on Academic Integrity ([njit.edu/academics/integrity.php](http://njit.edu/academics/integrity.php)). It describes infractions of academic integrity and penalties for violations, including, for the most serious violations, an XF grade in the course or expulsion. All work that you represent as your own must, in fact, be your own. Work done by others must be given proper credit.

Students will be informed of any modifications of the syllabus during the semester.
CS 113 - Introduction to Computer Science I
Course Syllabus, Fall 2015

Format

The format has lecture and lab components (recitations):
- One lecture plus one recitation per week
- All students attend the same lecture class
- Students partitioned into smaller groups attend recitation

Lectures + Recitation

Lecture Class Meeting Time: Thursday: 4:00 pm – 5:25 pm
Room: Kupfrian 118
Instructor: Dr. Narain Gehani

There are 3 recitation sections attached to the above lecture class. You will be in one of them:

- Recitation Section 001: Monday: 1:00 pm – 2:25 pm
  Room: GITC 2115C (Dr. Michael Baltrush)
- Recitation Section 003: Tuesday: 11:30 am – 12:55 pm
  Room: GITC 2315C (Dr. Michael Baltrush)
- Recitation Section 006: Wednesday: 8:30 am – 9:55 am
  Room: GITC 2400 (Ms. Junilda Spirollari)

Contact Information

Instructor: Dr. Narain Gehani
Office: GITC 4304
Email: gehani@njit.edu
Office Hours: Thursday: 10:30 am – 12:00 pm

TA Contact Information

TA: TBA
Office: TBA
Email: TBA
Office Hours: TBA
Overview

This course is a comprehensive introduction to the Java programming language teaching writing, testing and debugging of programs. Topics covered fundamental Java object-oriented programming. Topics covered are primitive data types, variables, assignments expressions and operators, control statements, recursion, design and use of classes, arrays, and I/O. Other topics covered are testing and debugging and writing programs that work reliably. The course guides students to the development of comprehensive Java applications.

Textbook


Please ignore the graphics portions at the end of each chapter. Graphics are not part of this course. Similar is the case with applets.

Reference Material

Java’s strength comes from the large number of libraries. Language is relative simple – but made complex from the large library facilities. For details of Java library facilities, please refer to

/docs.oracle.com/javase/8/docs/api/overview-summary.html

Programming Environment

We will use jGRASP, an integrated development environment (IDE) that supports Java, for program development. This environment is supported at NJIT’s lab classrooms.

To acquire this environment for personal use, you can download it from

/www.jgrasp.org

You will also need to install Java development kit (JDK). Follow Java (JDK) download instructions on the jGRASP download page – go to Oracle website and download Java.

/www.oracle.com/technetwork/java/javase/downloads

You can also develop and run Java programs by accessing the Java compiler (javac) and interpreter (java) directly from the Windows Command Window. You need to edit Windows Environment variables PATH to point to the BIN directory in the Java software folder.
Prerequisites

CS100 – Roadmap to Computing or equivalent

Course Policies

- Attendance is mandatory.
- Moodle (moodle.njit.edu) will be used for course communication. Please keep checking Moodle.
- **Homework assignments must be submitted in hard copy.**
- Homework assignments will not be accepted late except for special circumstances (such as jury duty or medical problem), for which you must provide documentation.
- All submitted work (including exams) must include your name and student ID.
- Plagiarism will result in zero credit for the assignment and/or an XF grade in the course.
- Cell phones must be turned off during class.
- Students will be informed of any modifications of the syllabus during the semester.

Material to be Covered

- Introduction to programming and Java programming language
- Data and Expressions
- Using Classes and Methods
- Decisions and Loops
- Arrays and I/O
- Objects-Oriented Programming
  a. Object-Oriented Design
  b. Defining Classes and creating Objects
  c. Defining methods
  d. Inheritance
- Testing and Debugging
- Exceptions
- Recursion

Course Goals

Upon completing the course, the students would accomplished the following:

1. Learn how to use core Java facilities with a focus on problem solving
2. Learn how to define classes (objects) and use them to write programs
3. Be able to write non-trivial Java programs.
Learning Outcomes

Students will be able to work with a customer (someone who wants a software/program to be developed) to understand the problem to be solved, design and write the (Java) program working by themselves.

Specifically, having determined the program to be written, students will be able to write Java programs that include programming concepts / Java facilities declarations, conditional and loop statements, define classes, methods (including recursive methods), handle errors, and debug/test programs for correct behavior.

Students will also be prepared with Java programming skills (as above) for advanced courses.

What do the Assignments Accomplish?

By doing assignments,
1. a student will develop problem solving expertise,
2. write Java programs putting to practice the programming concepts / Java facilities learned,
3. learn object-oriented programming, and
4. learn about software requirements and testing.

Writing Java programs will involve the use of declarations, conditional and loop statements, define classes (including inheritance), methods (including recursive methods), handle errors, and debugging / testing programs for correct behavior.

Performance

Assignments and the exams aim to assess
1. Java programming skills in the context of the use of Java facilities (as mentioned above) to solve problems, and
2. understanding of programming concepts.

Evaluation

Evaluation (final grade) will be based on the following items:

- Attendance 5%
- Homework 30%
- Midterm Exam 32%
- Final Exam 33%

Please note that scores entered on Moodle are the raw scores for each individual item – not allocated as per the above percentages.
Exam Policies

You must bring a photo ID to all exams. Students with special needs are advised to make arrangements with Disability Services.

Only one midterm.

There are no makeup exams. If you miss a midterm because of a documented special circumstance you may receive an imputed grade based on the final exam.

If you believe that you deserve more credit than you have been awarded on a particular exam problem, you may request, within 48 hours of the exam being returned, that it be regraded. Your entire exam will be regraded, which may result in points being added or subtracted.

Exams do not require any portable electronic devices, such as cell phones or calculators, and all such devices must be put away and turned off during the exam. Cell phones must be on silent and cannot be answered during the exam.

University Code on Academic Integrity

Read the University Code on Academic Integrity (njit.edu/academics/integrity.php). All work that you represent as your own must be your own. Work done by others must be given proper credit.

Tentative Weekly Coverage of Material

The following table shows approximately how much time may be devoted to each topic. Actual class lectures may vary in pace and order. Recitations will supplement lectures. Students should also supplement learning by reading in the book topics covered in class. A reading list is given following the table:

<table>
<thead>
<tr>
<th>Week (Approx)</th>
<th>Lecture</th>
<th>Recitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to programming and Java programming language</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Basics of Java programs</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Software Development using Stepwise Refinement</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Nuts &amp; Bolts of Java programs ...</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Nuts &amp; Bolts of Java programs ...</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>More Software Development using Stepwise Refinement</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Printing output</td>
<td></td>
</tr>
<tr>
<td>6-7</td>
<td>Java Statements</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Strings</td>
<td></td>
</tr>
<tr>
<td>Chapter</td>
<td>Topic</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>More Software Development using Stepwise Refinement</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Arrays + Program Development using Stepwise Refinement</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Object-Oriented Programming</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Object-Oriented Programming</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Printing Output – details about printf</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Testing &amp; Debugging + Passing Arguments</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Data structures</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>More about classes – Inheritance, Abstract Classes</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Recursion, Searching, Sorting</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Exceptions</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>More Data Structures</td>
<td></td>
</tr>
</tbody>
</table>

**Some Important Dates**

**Exam Dates**

<table>
<thead>
<tr>
<th>Exam</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final</td>
<td>TBD by the Registrar</td>
</tr>
</tbody>
</table>

**Assignment Due Dates**

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Due date is your first recitation after the date listed below. Hand in and pick up graded assignment in recitations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>September 10</td>
</tr>
<tr>
<td>2</td>
<td>September 24</td>
</tr>
<tr>
<td>3</td>
<td>October 8</td>
</tr>
<tr>
<td>4</td>
<td>October 22</td>
</tr>
<tr>
<td>5</td>
<td>November 5</td>
</tr>
<tr>
<td>6</td>
<td>November 19</td>
</tr>
<tr>
<td>7</td>
<td>December 3</td>
</tr>
</tbody>
</table>
Readings From
Java Software Solutions by Lewis & Loftus – 8/E
as Supplement for Topics Covered in Lectures

Java Programming Language: Pages 26 – 36.
Program Development: Pages 36 – 44.
Object-Oriented Programming: Pages 44 – 49.
Strings: Pages 58 – 65.
Variables + Assignment: Pages 65 – 70.
Primitive Data Types: 71 – 75.
Expressions: Pages 75 – 83.
Data Conversion: Pages 83 – 87.
Interactive Programs: Pages 87 – 92.
Creating objects: Pages 114 – 118
String Class: Pages 118 – 122
Random & Math Classes: 126 – 132
Wrapper Classes: 141 – 144
Class Dic: Pages 162 – 167
Encapsulation + Visibility Modifiers: Pages 169 – 172
Methods (includes constructors): Pages 172 – 182
Boolean expressions & if statements: Pages 210 – 229
Loops: Pages 230 – 241
Switch statement & conditional operator: Pages 270 – 275
Do statement & for loops: Pages 275 – 284
static variables & methods: Pages 305 – 309
Interfaces: Pages 322 – 327
Method Parameters & Overloading: Pages 338 – 345
Testing: Pages 345 – 349
Arrays: Pages 380 – 392
Arrays of Objects: 392 – 401
Command-line Arguments: Pages 402 – 404
Variable Length Parameter Lists: Pages 404 – 408
Two Dimensional Arrays: Pages 408 – 412
Inheritance: Pages 443 – 453
Sorting: Pages 504 – 513
Searching: Pages 513 – 519
Exceptions: Pages 537 – 548
Recursion: Pages 584 – 601
Queues & Stacks: Pages 627 – 631
Prerequisites

CS 113 or completion of a required 100 level GUR course in CS plus an approved CIS105. You must be able to program in Java.

Course Goals

This course is an introduction to the study of data structures and algorithms, emphasizing implementations in the Java programming language. At the conclusion of the course, students will be able to write computer programs using standard data structures and algorithms, and be able to bound the resources used by an algorithm.

Learning Outcomes

- The ability to write computer programs using standard data structures and algorithms.
- The ability to bound the resources used by an algorithm.

Textbooks


For information on the book, see the author’s web page:

http://people.cs.vt.edu/~shaffer/Book/

You may find the following textbook useful:


Course Materials and Communications

We will be using the Moodle system (http://moodle.njit.edu). All class information (including this syllabus, class notes, homework assignments) will be posted there. You can post questions (and answers) there, and I will post occasional updates. All communications should be through the “News forum” (do not use “Messages”).

If you have a personal issue that you wish to bring to my attention (for example if you need to inform me that you need to miss class due to illness) you should email or call me, or speak to me in person. For other communications, you should use Moodle (for example, questions on homework or what will be on the exam).

We will also be using AFS, so make sure that your account is working. You should have a subdirectory for CS114. The homework for each week will be posted on Moodle.

At the start of each week I will post class notes and homework assignment for the coming week.
Grading

Homework will be assigned each week in the recitation section. Typically, homework assignments will have a programming component and an analysis component. There will be lab exercises (typically involving programming) based on each previous week's homework. You must use the lab computers to complete these in-class assignments.

There will be an in-class midterm on Friday, October 16. There will be a final exam, at a time that will be announced by the registrar later in the semester. The course grade will be based on the final (35%), midterm (20%) and lab exercises and homework (45%).

If you are unable to meet any of the course requirements (for example due to illness), you must contact me immediately (email or leave a phone message). In order to be excused from a component of the course that contributes to the final grade, you must supply documentation explaining your absence to the office of the dean of students, and they will in turn contact me.

Academic Honesty

It is every student's responsibility to understand and adhere to the provisions of the academic honor code. You may discuss homework problems with your colleagues, but all written work must be your own. Copying programs or written assignments from any source is a serious violation of the academic honor code. Violations of the exam instructions will result in a score of zero for the exam. Any evidence of dishonesty will be reported to the Dean of Students for disciplinary action, and will result in a grade of zero for the course component involved.

Tentative Course Outline

Week   Topics
1. Introduction
2. Recursion
3. Math background, common functions
4. Algorithm analysis, asymptotic analysis
5. Lists
6. Stacks, queues
7. Dictionaries
8. Binary trees, search trees
9. Priority queues, heaps
10. Sorting
11. Sorting lower bound, linear-time sorting
12. Selection
13. Graphs
14. Graph algorithms
CS 241 COURSE SYLLABUS - FALL 2015

NJIT ACADEMIC INTEGRITY CODE: All Students should be aware that the Department of Mathematical Sciences takes the University Code on Academic Integrity at NJIT very seriously and enforces it strictly. This means that there must not be any forms of plagiarism, i.e., copying of homework, class projects, or lab assignments, or any form of cheating in quizzes and exams. Under the University Code on Academic Integrity, students are obligated to report any such activities to the Instructor.

CS 241: Foundations of Computer Science I (Discrete Mathematics for CS)

Number of Credits: 3

Course Description: This course provides the mathematical and analytical foundations of computer science and its applications to various areas in CS. The course covers the material traditionally known as "discrete mathematics", with special emphasis on CS applications and an analysis of algorithms. The course topics include sets and logic, proof techniques, proof by induction, functions and relations, analysis of algorithms, recursion, recurrence equations, divide-and-conquer design technique, counting methods (permutations and combinations), basic discrete probability, and if time permits, introduction to number theory, and a brief introduction to graphs and trees. Prerequisites: Prerequisites: CS 114: Intro to Computer Science; Math 112: Calculus II.

Course Objectives (what you are expected to know to complete this course)

1. Know basic mathematical tools and terminologies used in computer science
2. Know set algebra, propositional logic, reasoning, and basic proof techniques
3. Know induction, recursion, recurrence equations, and how they are interrelated
4. Know the mathematical tools used to analyze efficiency of algorithms
5. Implement simple programs and run experiments to measure their time complexity
6. Learn permutations/combinations, basic discrete probability and applications


Instructor: CS 241-001 Adrian Ionescu

Grading Policy: The final grade in this course will be determined as follows:

- Quizzes, Homework, Projects: 30%
- Midterm Exam: 30%
- Final Exam: 40%
Drop Date: Please note that the University Drop Date November 2, 2015 deadline will be strictly enforced.

Homework Policy: Homework problems will be assigned in class.

Attendance: Attendance at all classes will be recorded and is mandatory. Please make sure you read and fully understand the Department’s Attendance Policy. This policy will be strictly enforced. Absences from class will inhibit your ability to fully participate in class discussions and problem solving sessions and, therefore, affect your grade. Tardiness to class is very disruptive to the instructor and students and will not be tolerated. Each student should have contact information of several fellow students to get homework assignments and class notes when absent. You are responsible for everything that happens in class whether you are present or not.

Makeup Exam Policy: There will be NO MAKE-UP EXAMS during the semester. In the event the Final Exam is not taken, under rare circumstances where the student has a legitimate reason for missing the final exam, a makeup exam will be administered by the math department. In any case the student must notify the CS Department Office and the Instructor that the exam will be missed and present written verifiable proof of the reason for missing the exam, e.g., a doctor’s note, police report, court notice, etc., clearly stating the date AND time of the mitigating problem.

Further Assistance: For further questions, students should contact their Instructor. All Instructors have regular office hours during the week. These office hours are listed at the link above by clicking on the Instructor's name.

Cellular Phones: All cellular phones and beepers must be switched off during class.

<table>
<thead>
<tr>
<th>Date, 2015</th>
<th>Day</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 1</td>
<td>T</td>
<td>First Day of Classes</td>
</tr>
<tr>
<td>September 7</td>
<td>M</td>
<td>Labor Day - University Closed</td>
</tr>
<tr>
<td>September 8</td>
<td>T</td>
<td>Monday Classes Meet</td>
</tr>
<tr>
<td>November 2</td>
<td>M</td>
<td>Last Day to Withdraw from a Class</td>
</tr>
<tr>
<td>November 25</td>
<td>W</td>
<td>Classes follow a Friday Schedule</td>
</tr>
<tr>
<td>November 26</td>
<td>R</td>
<td>Thanksgiving Recess Begins</td>
</tr>
<tr>
<td>December 11-14</td>
<td>F&amp;M</td>
<td>Reading Days</td>
</tr>
<tr>
<td>December 15-21</td>
<td>M-S</td>
<td>Final Exams</td>
</tr>
</tbody>
</table>
## COURSE OUTLINE

<table>
<thead>
<tr>
<th>WEEK</th>
<th>Johnsonbaugh</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Chapter 1</td>
<td><strong>Sets and Logic</strong>: Sets, Propositional Logic, Quantifiers</td>
</tr>
<tr>
<td>3-4</td>
<td>Chapter 2</td>
<td><strong>Proof Techniques</strong>: Direct Proof, Counterexample, Contrapositive, Proof by Contradiction, Enumeration Proof, Proof by Induction; Strong Induction</td>
</tr>
<tr>
<td>5-6</td>
<td>Chapter 3</td>
<td><strong>Functions and Relations</strong>: Functions Relations Properties: Reflexive, Symmetric, Transitive; Partial Order, Total Order, Equivalence Relations, Matrices of Relations Application: Relational Databases</td>
</tr>
<tr>
<td>7</td>
<td>Chapter 4</td>
<td><strong>Algorithms</strong>: Analysis of Algorithms, Recursive Algorithms, Use of Recurrences to Analyze Algorithms</td>
</tr>
<tr>
<td>8</td>
<td>Midterm</td>
<td><strong>REVIEW and MIDTERM</strong></td>
</tr>
<tr>
<td>9-10</td>
<td>Chapter 7</td>
<td><strong>Recurrence Equations</strong>: Divide-and-Conquer Recurrences, Master Theorem, Linear Recurrences</td>
</tr>
<tr>
<td>11-12</td>
<td>Chapter 6</td>
<td><strong>Counting Methods</strong>: Permutations and Combinations, Principle of Inclusion/Exclusion, Pigeonhole Principle, Introduction to Basic Probability</td>
</tr>
<tr>
<td>13</td>
<td>Chapter 5</td>
<td><strong>Introduction to Number Theory</strong> (if time permits)</td>
</tr>
<tr>
<td>14</td>
<td>Chapters 8-9</td>
<td><strong>Introduction to Trees and Graphs</strong></td>
</tr>
<tr>
<td>15</td>
<td>Review</td>
<td><strong>REVIEW FOR FINAL EXAM</strong></td>
</tr>
</tbody>
</table>

---

**Prepared By:** Prof. Adrian Ionescu

**Last revised:** September 1, 2015
CS 114H: Honors Introduction To Computer Science II

Syllabus

Instructor Info

Instructor: Jonathan Kapleau
Office: GITC 4212
Office Phone: 973-596-2657

Course Description

Fundamentals of computer science are introduced, with emphasis on programming methodology and problem solving. Topics include basic concepts of computer systems, software engineering, algorithm design, programming languages and data abstraction, with applications. A high level language is fully discussed and serves as the vehicle to illustrate many of the concepts. Java is used in this course.

Learning Objectives

- The student will be able to write computer programs using standard data structures and algorithms.
- The student will be able to bound the resources used by an algorithm.

Textbooks

Data Abstraction and Problem Solving With Java: Walls and Mirrors
Frank M. Carrano, Janet J. Prichard
ISBN: 0-321-30428-4

Grading Scheme

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Final</td>
<td>30%</td>
</tr>
<tr>
<td>Projects</td>
<td>40%</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10%</td>
</tr>
</tbody>
</table>

Topics
• Recursion
• Data Abstraction
• Linked Lists
• More Recursion
• Stacks
• Queues
• Algorithm Efficiency & Sorting
• Trees
• Tables & Priority Queues
• More Trees
• Graphs

Attendance Policy

Attendance in every lecture is mandatory. If a student is absent from lecture five times (the first day counts), the student's name will be recommended for withdrawal to the Dean of Freshman Studies. Two lates is equivalent to one absence. Make sure that you fully understand this attendance policy.

Cheating Policy

Cheating on a programming assignment results in zero credit for all students involved. Programming assignments may NOT be solved in collaboration, unless specifically stated in the assignment. Cheating on an exam will result in an "F" in the course.

You may discuss problems with each other. Where does discussion end and cheating start? You may NOT copy lines of code from anybody or anywhere. You may NOT use code in your assignments that you did not write. As a general rule: If you don't understand the code and can't explain the code, you can't use the code.

Please familiarize yourself with the NJIT Honor Code. Violations of the honor code will be dealt with seriously and reported immediately to the Dean of Students.

Late Policy

To receive full credit all programming assignments must be handed in on time. Assignments that are not submitted on time will be penalized for each day that they are late. The type and severity of the penalty will be determined by the assignment. In general, a deduction of 10 points (out of 100) for the first day, 20 additional points for the second day, 30 additional points for the third day, and 40 additional points for the fourth day late will be subtracted from the final grade of the assignment.

Prerequisites

This course requires that you have an understanding of the basics of Java as well as the basics of Computer Science. These are things that you should have learned in CIS 113 or similar.

There are "absolute" and "recommended" prerequisites. Ignorance of any "absolute" prerequisite will
most likely result in an "F". These topics will not be covered in class as it is assumed that you are already familiar with them. The "recommended" prerequisites may be quickly reviewed in class, but you should already have some familiarity with these topics. If you are unfamiliar with these topics you will most likely have a difficult time understanding the covered material.

Absolute Prerequisites:

- Chapter 1 of the textbook.
- basics of editing, compiling, and executing Java programs.
- elementary data types and common operations on them.
- basic input & output
- all control structures including switch and do...while
- method declarations and definitions.
- one-dimensional arrays.
- constants and variable declarations.
- good style.
- the concept of an algorithm.

Recommended Prerequisites:

- classes.
- two-dimensional arrays.
- enumerations.
- simple recursion.
CS252 Computer Organization and Architecture
Fall 2015

Instructor: Dr. Michael A. Baltrush

Office: GITC 4310
Phone: 973.596.3386
E-mail: baltrush@njit.edu
Home Page: Moodle for course


Class: CS252-001 TR 1:00-2:25, Kupf 204
CS252-003 TR 8:30-9:55, Kupf 104

Office Hours:
Monday 2:30-3:10 PM
Tuesday 2:30-3:10 PM
Thursday 2:30-3:30 PM
Other times: By appointment

Exams are closed book and notes.

Final Grade:
1/3 In class exams 1-1/2 hours (tentative dates)
   Exam1: October 13, 2015
   Exam2: November 19, 2015
1/3 Final exam is cumulative: Scheduled by Registrar
1/3 Collected and graded homework and assembly programs

Please be aware that the semester is not over until the after the final exam period. Do not schedule trips during the final exam period.

If you don’t understand something from class, ASK a question.

All students must attend on the stated exam date, time and place. If you have another commitment you must make arrangements for that. The only valid reasons for missing the exam are accident or sickness. If you do miss the exam due to unforeseen circumstances you should supply the Dean of Students office with supporting documentation (Doctor’s note, copy of the police report) so a makeup exam can be scheduled.

You will need a pencil and eraser to take the exams.

You may NOT leave the room during the exam.
Please turn off all cell phones and pagers. The University Code on Academic Integrity (referenced at the URL) [http://www.njit.edu/academics/pdf/academic-integrity-code.pdf](http://www.njit.edu/academics/pdf/academic-integrity-code.pdf) in enforced in this class.

Homework from the text is assigned in class and posted in Moodle at the completion of a chapter and is due one week after assignment. In Moodle terms this is an 'offline activity' and as such a hardcopy must be handed in by the due date. It will be collected, graded and returned. Late work will not be accepted. There is no make-up homework. (Late defined as being after the class where the work is reviewed.) Programming assignments using the ARM assembly language (approximately 5) will be available in Moodle as Word documents.

Topics are covered in chapter order:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Topic</th>
<th>Subtopics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1</td>
<td>Basic Structure of Computers</td>
<td>Two’s Complement representation, integers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Floating Point</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Machine Instructions and programs</td>
<td>Operand Addressing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Machine Instructions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subroutine</td>
</tr>
<tr>
<td>Appendix D</td>
<td>ARM Instruction Set and Examples</td>
<td>ARM Instruction set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARM IDE</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Input/Output Organization</td>
<td>Programmed (Polling)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software for above</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Buses, Ports</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Basic Processing Unit</td>
<td>Instruction execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hardware Components</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control signals</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hardwired/Microprogram Control</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Pipelining</td>
<td>Basics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Various Hazards</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Input/output Organization</td>
<td>Bus structure/operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interfaces</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interconnection standards</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Memory system</td>
<td>Semiconductor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROM’s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Direct Memory Access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cache</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Virtual memory</td>
</tr>
<tr>
<td>Chapter 9</td>
<td>Arithmetic</td>
<td>Integer Adders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multiply Integers</td>
</tr>
</tbody>
</table>
If you do not attend the class when the work is assigned check with one of your classmates or on the Moodle site.

You will need to load an ARM academic package on your computer to do the assembly language programming exercises. For Windows 7 use the tool-chain version originally from CodeSourcery. This will also load under Windows 8 although you must use the console interaction to install the software. It will indicate an error condition on completion since the installer cannot put shortcuts on the start screen. You will need to load the ARM emulator from Moodle to execute the resulting program set.

Attendance is taken during the class period. If your name does not appear on the roster please see the Registrar.
CS 280: Programming Language Concepts
Syllabus, Fall 2015

Professor Gerard Ryan
GITC 4303
973.642.4029
gerard.w.ryan@njit.edu
gwryan@njit.edu
http://web.njit.edu/~gwryan
Twitter: @gwryanNJIT

Fall 2015 Office Hours: Monday 10:00-11:30, Thursday 5:00-6:00, or by appointment

CS 280 - Programming Language Concepts
Conceptual study of programming language syntax, semantics and implementation. Course covers language definition structure, data types and structures, control structures and data flow, run-time consideration, and interpretative languages.

Please include CS280 and your section number in the Subject: line of any email you send; it will make it easier to manage my emails. I will do the same in emails I send to you.

Course Webpage: http://web.njit.edu/~gwryan/CS280
All course materials, including lecture notes, assignments and solutions, will be posted on the web page. Announcements and notices will also be posted. WHEN IN DOUBT, CHECK THE WEBSITE. Moodle will be used for submissions and online Q&A.

Textbook:

Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance/Other</td>
<td>7%</td>
</tr>
<tr>
<td>Programs</td>
<td>40%</td>
</tr>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Final</td>
<td>33%</td>
</tr>
</tbody>
</table>

A student who does not submit two or more programming assignments will receive an F for this course.
Topics:

- Common features of programming languages
- Lexical Syntax
- Grammars
- Names
- Types
- Semantics
- Expressions
- Control Flow
- Subprograms
- Encapsulation
- Memory Management
- Event Handling
- Concurrency

Goals for the Course:

The student will be able to recognize similar features of different programming languages.

The student will have an easier time learning new programming languages.

The student will gain an appreciation of the strengths and weaknesses of different programming languages.

The student will demonstrate an ability to apply knowledge of computing and mathematics appropriate to the discipline.

The student will demonstrate an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

The student will demonstrate an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

The students will recognize the need for and an ability to engage in continuing professional development.

The student will demonstrate an ability to use current techniques, skills, and tools necessary for computing practice.
Ethical Conduct (This should go without saying, people. Seriously.)

Cheating during in-class tests or take-home examinations or homework is, of course, illegal and immoral. Programming assignments are NOT collaborative efforts. You may discuss problems with each other, but you may NOT copy lines of code from anybody or anywhere. You may NOT use code in your assignments that you did not write.

The essential quality of the NJIT University Code on Academic Integrity is that each student shall demonstrate honesty and integrity in the completion of all assignments and in the participation of the learning process. Adherence to the University Code on Academic Integrity promotes the level of integrity required within the university and professional communities and assures students that their work is being judged fairly with the work of others. See http://www.njit.edu/academics/pdf/academic-integrity-code.pdf

Notes on Programming Projects:

- All programming projects must be submitted electronically via Moodle
  - Put SOURCE CODE ONLY into a SINGLE zip file
  - Upload the zip file into Moodle.
  - You can resubmit as often as you like.
  - Do not wait until the last minute.
  - Late submissions are penalized

- A printout of your project DOES NOT NEED TO BE SUBMITTED

- The assignments will be in C++

- Every programming project should include a block of identifying comments at the very top of each file, as follows:
  - CS 280, the section number, and SPRING 2015
  - Your name
  - Assignment #

- Your code should include comments indicating how you solved the problem

- Programming projects will be graded on a scale of zero to 10

- Each day or part of a day that you are late reduces your grade by 1 point. Online submissions will be disabled five days past the due date.

- If your program does not compile, your grade will be a 1

- Partial answers or incorrect output will reduce your grade

- A student who does not submit two or more programming assignments will receive an F for this course.
Course Outline:

- Introduction [chapter 1 + 2]
- Lexical and Syntax Analysis [chapter 3 & 4]

- Names [chapter 5]
- Types [chapter 6]
- Expressions [chapter 7]
- Statements + Subprograms [chapter 8-10]

- Encapsulation and ADTs [chapter 11]
- Object Oriented Programming [Chapter 12]
- Memory Management
- Events
- Topics Survey
- FINAL EXAM

Note that exams are cumulative.

<table>
<thead>
<tr>
<th>Important Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-Sep</td>
</tr>
<tr>
<td>23-Sep</td>
</tr>
<tr>
<td>19-Oct</td>
</tr>
<tr>
<td>26-Oct</td>
</tr>
<tr>
<td>11-Nov</td>
</tr>
<tr>
<td>9-Dec</td>
</tr>
<tr>
<td>15-Dec-21-Dec</td>
</tr>
<tr>
<td>CLASS MEETS</td>
</tr>
<tr>
<td>Program 1 Due</td>
</tr>
<tr>
<td>Program 2 Due</td>
</tr>
<tr>
<td>Midterm</td>
</tr>
<tr>
<td>Program 3 Due</td>
</tr>
<tr>
<td>Program 4 Due</td>
</tr>
<tr>
<td>FINALS WEEK</td>
</tr>
</tbody>
</table>
Fall 2015 CS288 Intensive Programming in Linux - Syllabus

Check the announcements regularly.

- Class Web page: http://www.cs.njit.edu/~sohna/cs288
- Instructor: Andrew Sohn, GITC 4209, (973)596-2315, email: sohn_at_cs_dot_njit_dot_edu
- Office Hours: Tue 11:30am-1pm, Thur 5:30-7pm
- Teaching assistant: no TA, NJIT policy
- Grader: TBA if I get one
- Class time and location: TR 100pm-225pm GITC 1100, See the registrar's page http://www.njit.edu/registrar/schedules/courses/gall/2015F.CS.html
- Platform: Linux, distro Fedora 20 or above, multi-booted on bare-metal recommended; Do not use virtual machine for daily interactive computing.
- Tools: Bash, C, Python, mySql, DOM, and PHP.
- Grading: assessment quiz on Day 1 (2%); weekly programming assignments (0%); Test 1, Tue, 9/29/2015 (30%); Test 2, Tue, 10/27/2015 (30%); Final exam (38%) - See the registrar's page for date, time and location. Exam questions will be derived directly from programming assignments. Do your homework from scratch. Be prepared to spend two hours a day on homework.
- NJIT policy on missed exams: There will be no make up exam(s). You must plan your semester accordingly, especially if you work. Should you miss exam(s) due to emergency, (a) go to the Dean of students, (b) explain your situation as to why you had to miss, and (c) ask to issue a memo to me. If and when I receive a memo from the Dean ib your missed exam, I will copy your next exam score to the missing one. Those who miss final exam will fail in the course unless you demonstrate a true emergency again through the office of the Dean of students. This is the NJIT policy for missed exams. No other policy will be applied. No exceptions will be made.
- NJIT policy on video recording class materials: You may not video record the class materials. You may not put any video/audio recorded class materials on the Web/Internet. It is against the Univeristy policy.
- Disagreement with exam marking/scores: If you disagree with your exam scores/marks, you may dispute within a week of receiving/seeing the graded exam paper. After a week, no exams will be contested.
- Grading dispute: If you disagree with your grade, you may contest after the first day but within a week of the following semester. After a week of the first day of the following semester, no grading dispute will be considered.
- Exam papers: Exam papers will not be returned to you. They have to be kept for accreditation purposes. You may come and view your graded exam paper in my office during the office hours.
- See Academic Integrity
Lecture schedule - Contents may change according to class pace

1. **STAGE 1 - learning the most basic and fundamental knowledge**
   Intro to Linux, LAMP (Linux, Apache, mySql(maria), Python/Php), virtualization
2. Intro to Bash shell scripting - variables, assignments
3. Intro to Bash shell scripting continues - arrays, lists, functions
4. Recursive directory traversal in Bash - depth first and breadth first
5. Pattern matching with regular expression (grep)
6. Introduction to C pointers, ref/dereferences,
7. Pointers to pointers, array of pointers, function pointers
8. Malloc/free and basic structure handling with simple linked list
9. Structure handling - swap and push, structure handling with multiple links
10. **STAGE 2 - tools for building an end-to-end realworld application**
    Sorting - fast integer radix sort for integers
11. Sorting - introduction to floating point representation
12. Sorting - fast radix sort for floats
13. State space search - depth first, breadth first search
14. State space search - heuristic-based intelligent search
15. State space search - intelligent bi-directional search
16. Matrix computation: a system of linear equation solvers
17. Matrix computation: introduction to iterative methods
18. Matrix computation: application to spectral graph partitioning
19. **STAGE 3 - an end-to-end realworld application**
20. Web processing - fetching with wget using Bash scripting, intro to DOM tree, properties, methods
21. Web processing - DOM tree navigation, data extraction using Python minidom
22. Web processing - getting up and running mySql server, mySql DB construction, data injection
23. Web processing - getting up and running Apache server, reading DB using PHP, constructing clickable/sortable, plotting charts and graphs and presenting data using PHP
24. **STAGE 4 - extending tools and applications to run on many-core machines**
25. Introduction to multicore/parallel computing using MPI (optional OpenMP) - point to point communication
26. Introduction to MPI (optional OpenMP) - collective group communication
27. Simple matrix computation for multicore/multiple machines using MPI (optional parallel radix sort using MPI on a cluster of machines)
CS 341: Foundations of Computer Science II
Fall 2015, Face-to-Face Section

Course Info

Class Times: Wednesday, Friday, 1:00 – 2:25
Instructor: Prof. Marvin K. Nakayama
Office: GITC 4312
Phone: 973-596-3398
E-mail: mailto:marvin@njit.edu
Office Hours: Wednesday, Friday, 11:00am – 12:00pm, or by appointment.
Course Webpage: http://web.njit.edu/~marvin/cs341

Description

This course presents some of the most fundamental results in theoretical Computer Science. These results attempt to answer, in a precise mathematical sense, the following two questions, which are of practical as well as philosophical interest:

1. Can a given problem be solved by computation?

2. How efficiently can a given problem be solved by computation?

We often focus on problems rather than on specific algorithms for solving problems. To answer both questions mathematically, we will need to formalize the notion of “computer” or “machine.” The course outline breaks naturally into three parts:

1. Models of computation (Automata Theory)
   - Finite automata
   - Push-down automata
   - Turing machines

2. What can we compute? (Computability Theory)

3. How efficiently can we compute? (Complexity Theory)

Specifically, the topics covered will include regular languages (finite automata, regular expressions), nonregular languages, context-free languages (context-free grammars, pushdown automata), non-context-free languages, Turing machines and variants, Church-Turing Thesis, undecidability, reducibility, time complexity, and complexity classes P, NP, and NP-complete.

Student Course Outcomes

The student course outcomes of the course are to

- Understand the capabilities and limitations of different models of computation.
• Understand what problems can be solved by computation and which cannot.

• Understand what problems can be solved efficiently and those for which there is no known efficient solution.

The specific learning objectives are that after completing the course, students will be able to

• Classify a particular language as regular, context-free, decidable, Turing-recognizable or non-Turing-recognizable.

• Provide a finite automaton and regular expression for a regular language.

• Prove that a nonregular language is not regular.

• Provide a context-free grammar and pushdown automaton for a context-free language.

• Prove that a non-context-free language is not context-free.

• Provide a description of a Turing machine for a decidable language.

• Prove or disprove closure properties (under union, intersection, complementation, Kleene star) of classes of languages.

• Prove that certain languages are undecidable or non-Turing-recognizable.

• Understand nondeterminism and its role in computation and complexity theory.

• Understand the significance of complexity classes P, NP and NP-complete, and carry out some NP-completeness reductions.

Textbook


Prerequisites

Before taking CS 341, you must complete all of the following with grades of C or better:

1. A 100-series general undergraduate required course in CS

2. CS 241 (Foundations of Computer Science I)

3. CS 280 (Programming Language Concepts).
Grading

Your course grade will be determined by two projects (programming assignments), two in-class midterms and a final exam. **All exams will be closed book and closed notes.** The midterm exams will be 85 minutes long, and the final exam will be 2.5 hours long. Unless notified otherwise, the dates of the midterms and the due date for the projects are as given in the schedule at the end of this document.

Your final grade will be based on the following weights:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (2)</td>
<td>20%</td>
</tr>
<tr>
<td>Midterms (2)</td>
<td>45%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>35%</td>
</tr>
</tbody>
</table>

**For each project,** students who do not turn in a minimally working program will get a 0 for the assignment and have their course grades at the end of the semester lowered by one step, e.g., from B to C+, or from C to D. Hence, if you do not turn in minimally working programs for two projects, your course grade will be lowered by two steps, e.g., from B to C or from C to F.

Course grades will be assigned on a curve using the following approach. First, I will rank everyone using the cumulative scores with the weights given above, and then assign preliminary grades based on that. The top group of students will get a preliminary grade of A, the next group will receive a preliminary grade of B, etc. Any student who scores less than 20 out of 100 on the final will automatically receive an F for the course.

After assigning preliminary grades, I will make adjustments for those who did not turn in minimally working programs. For each project for which you did not turn in a minimally working program, your preliminary grade will be lowered by one step. For example, if your preliminary grade was B and you only turned in one minimally working program out of the two projects, then your course grade is C+; if you did not turn in a minimally working program for neither project, then your course grade drops to a C. If you turned in minimally working programs for both projects, then your course grade is your preliminary grade based on the ranking of cumulative scores.

Course Policies

All of the course handouts (including lecture notes and assignments) are available in PDF format through my CS 341 homepage, whose address is given on the first page. **You must bring printouts of the lecture notes to each class.** To read the files, you will need to use a software package called Adobe Reader, which you can download for free using a link from the course web page. Be sure to check the course homepage each day since I will post announcements on it.

**As a general rule,** I do not give makeup exams, I do not allow students to take exams on alternate dates, nor do I allow students to turn in assignments late. Of course, if someone has a legitimate reason (e.g., jury duty, serious medical problem, conflict with a religious holiday), I will make allowances as long as you contact me beforehand (whenever it is feasible to do so) and provide proper documentation (e.g., a doctor’s note) to the Dean of Students, who will let me know if your absence can be excused. I will not accept excuses such as having too heavy a workload, having too many exams the same week, or simply forgetting.

If upon getting back graded material (e.g., exam, project) you think that you deserve more points on a particular part, I will regrade the entire thing. Thus, you may get more points on the one
part, but you may lose points on other parts. **You must ask for a regrade within one week of when the graded material is returned to the class or you are informed of your score, whichever is earlier.**

**There are no extra-credit assignments available.** So you need to make sure that you perform well on the assignments and exams.

For all exams, be sure to bring a photo ID. All exams will be closed book and closed notes. All portable electronic devices, such as cellphones and laptops, must be turned off during class. Students must check the course homepage and email each day. I will make announcements there. Students will be informed of any modifications or deviations from the syllabus throughout the course of the semester.

**moodle**

In this course we will use a computer-based group-communication system called moodle, which you can access at

http://moodle.njit.edu/

The website also provides instructions on using moodle, which requires logging in with your NJIT UCID. All projects must be submitted through moodle. Also, the course moodle page will have a link to video recordings of all of the lecture material. The preface of the lecture notes describes how the recorded modules correspond to the pages in the lecture notes.

**Homework Assignments**

All homework solutions are posted on the web at the beginning of the semester. **Do not turn in the homework assignments.** However, the only way you will learn the material is by doing the assignments, and many problems on the exams will be based on the homework problems.

When working on the homework problems, be sure to show all work and give reasons (e.g., proofs) for your answers. If your proof relies on a theorem or result from the book, be sure to either state the theorem number or page number from the book. Writing out complete solutions will help you prepare for the exams. Thus, it is important to do the homework, even though you do not turn them in.

**Projects**

The projects are mandatory, and must be turned by the due date/time through moodle. Late projects will be penalized at a rate of 25 points (out of 100) per 24-hour period. For example, since Programming Assignment 1 is due by 1:00pm NJ local time on 9/30/2015, you must submit all required documents for it by the due date/time to not be penalized. If you turn it in after the due date/time but up to 24 hours later, then you will automatically lose 25 points of the assignment. If you turn in a project between 24 hours and 48 hours late, then you will automatically lose 50 points of the project. If you turn in a project between 48 hours and 72 hours late, then you will
Automatically lose 75 points of the project. Projects completed over 72 hours late will not be accepted.

After the first two weeks of lectures, we will have covered enough material for you to do the first program. Expect to spend at least 5–10 hours on each project, so do not wait until the last minute to try to complete it.

All projects must be submitted through moodle:

http://moodle.njit.edu/

The website also provides instructions on using moodle.

**Academic Integrity**

Students must obey the University Code on Academic Integrity, which you can access from

http://www.njit.edu/education/pdf/academic-integrity-code.pdf

Any student caught violating this will be reported immediately to the Dean of Students. Violations include, but are not limited to,

- communicating with others during exams
- using unauthorized materials during exams
- copying/giving a computer program from/to another person.

For any exams that are less than 90 minutes in length, students will not be allowed to leave the classroom once the exam has begun.

**Schedule**

Unless I announce otherwise, the schedule for the semester is as below. Although you do not need to turn in the homework, you should complete the assignments according to the schedule below.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Reading</th>
<th>Homework</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro, Languages</td>
<td>Chapter 0</td>
<td>HW 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Regular Languages: DFA, NFA</td>
<td>Chapter 1</td>
<td>HW 2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>RL: Closure Properties, Reg Exp</td>
<td>Chapter 1</td>
<td>HW 3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Kleene's Thm, Nonregular Lang</td>
<td>Chapter 1</td>
<td>HW 4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CFG, PDA</td>
<td>Chapter 2</td>
<td>HW 5</td>
<td>Prog 1 due (9/30/2015)</td>
</tr>
<tr>
<td>6</td>
<td>CFG = PDA, Non-CFL</td>
<td>Chapter 2</td>
<td>HW 6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Turing Machines</td>
<td>Chapter 3</td>
<td>HW 7</td>
<td>Midterm 1 (10/14/2015)</td>
</tr>
<tr>
<td>8</td>
<td>Algorithms, Decidability</td>
<td>Chapter 4</td>
<td>HW 8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Decidability, Halting Problem</td>
<td>Chapter 4</td>
<td>HW 9</td>
<td>Prog 2 due (10/28/2015)</td>
</tr>
<tr>
<td>10</td>
<td>Undecidable Problems</td>
<td>Chapter 5</td>
<td>HW 10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Undecidability Reductions</td>
<td>Chapter 5</td>
<td>HW 11</td>
<td>Midterm 2 (11/11/2015)</td>
</tr>
<tr>
<td>12</td>
<td>Time Complexity, Class P</td>
<td>Chapter 7</td>
<td>HW 12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Classes NP, NP-Complete</td>
<td>Chapter 7</td>
<td>HW 13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>NP-Complete Reductions, Review</td>
<td>Chapter 7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CS 341: Foundations of Computer Science II
eLearning Section Syllabus, Fall 2015

Course Info

Instructor: Prof. Marvin K. Nakayama
Office: GITC 4312
Phone: 973-596-3398
E-mail: marvin@njit.edu (Be sure to include “CS341EL” in the subject line)
Only send e-mails about personal matters (e.g., your performance in the class).
Post any general questions on moodle (described below).
Office Hours: Wednesday, Friday, 11:00am - 12:00pm, or by appointment.
Course Webpage: http://web.njit.edu/~marvin/cs341

Description

This course presents some of the most fundamental results in theoretical Computer Science. These results attempt to answer, in a precise mathematical sense, the following two questions, which are of practical as well as philosophical interest:

1. Can a given problem be solved by computation?
2. How efficiently can a given problem be solved by computation?

We focus on problems rather than on specific algorithms for solving problems. To answer both questions mathematically, we will need to formalize the notion of “computer” or “machine.” The course outline breaks naturally into three parts:

1. Models of computation (Automata Theory)
   • Finite automata
   • Push-down automata
   • Turing machines

2. What can we compute? (Computability Theory)

3. How efficiently can we compute? (Complexity Theory)

Specifically, the topics covered will include regular languages (finite automata, regular expressions), nonregular languages, context-free languages (context-free grammars, pushdown automata), non-context-free languages, Turing machines and variants, Church-Turing Thesis, undecidability, reducibility, time complexity, and complexity classes P, NP, and NP-complete.
**Student Course Outcomes**

The student course outcomes of the course are to

- Understand the capabilities and limitations of different models of computation.
- Understand what problems can be solved by computation and which cannot.
- Understand what problems can be solved efficiently and those for which there is no known efficient solution.

The specific learning objectives are that after completing the course, students will be able to

- Classify a particular language as regular, context-free, decidable, Turing-recognizable or non-Turing-recognizable.
- Provide a finite automaton and regular expression for a regular language.
- Prove that a nonregular language is not regular.
- Provide a context-free grammar and pushdown automaton for a context-free language.
- Prove that a non-context-free language is not context-free.
- Provide a description of a Turing machine for a decidable language.
- Prove or disprove closure properties (under union, intersection, complementation, Kleene star) of classes of languages.
- Prove that certain languages are undecidable or non-Turing-recognizable.
- Understand nondeterminism and its role in computation and complexity theory.
- Understand the significance of complexity classes P, NP and NP-complete, and carry out some NP-completeness reductions.

**Textbook**


**Prerequisites**

*Before* taking CS 341, you must *complete* all of the following with grades of C or better:

1. A 100-series general undergraduate required course in CS
2. CS 241 (Foundations of Computer Science I)
3. CS 280 (Programming Language Concepts).
Grading

Your course grade will be determined by one homework assignment, three projects, one midterm and a final exam, which is cumulative. All exams will be closed book and closed notes. The midterm and final exams will be 2.5 hours long. Unless notified otherwise, the dates of the exams and the due dates for the projects are as given in the schedule at the end of this document. Your course grade will be determined by the following weighting scheme:

<table>
<thead>
<tr>
<th>HW EL</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (3)</td>
<td>30%</td>
</tr>
<tr>
<td>Midterm (1)</td>
<td>30%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>30%</td>
</tr>
</tbody>
</table>

For each project, students who do not turn in a minimally working program will get a 0 for the project and have their course grades at the end of the semester lowered by one step, e.g., from B to C+, or from C to D. Hence, if you do not turn in minimally working programs for two projects, your course grade will be lowered by two steps, e.g., from B to C or from C to F.

Course grades will be assigned on a curve using the following approach. First, I will rank everyone using the cumulative scores with the weights given above, and then assign preliminary grades based on that. The top group of students will get a preliminary grade of A, the next group will receive a preliminary grade of B, etc. Any student who scores less than 20 out of 100 on the final will automatically receive an F for the course.

After assigning preliminary grades, I will make adjustments for those who did not turn in minimally working programs. For each project for which you did not turn in a minimally working program, your preliminary grade will be lowered by one step. For example, if your preliminary grade was B and you only turned in one minimally working program out of the three projects, then your course grade is C+; if you did not turn in minimally working program out of the three projects, then your course grade drops to a C; if you did not turn in minimally working programs for two projects, then your course grade drops to a D. If you turned in minimally working programs for all projects, then your course grade is your preliminary grade based on the ranking of cumulative scores.

Exams

There will be one midterm, which will be given on a Saturday around the end of the 7th week of the semester; tentatively, the midterm date/time is Saturday, October 17, 2015. The final exam will be on Saturday, December 19, 2015. The times of the exams will likely be 9:30am – 12:00pm. More details of the exact dates, times and locations of the exams will be given during the semester.

If you live within a 75-mile radius of the NJIT Newark campus, you must take the midterm and final exam there on the date and time announced for each. If you live outside of the 75-mile radius of the NJIT Newark campus, then you need to nominate a proctor in Homework Assignment EL.

You must bring 2 forms of photo ID to all exams. All exams will be closed book and closed notes. Also, calculators will not be allowed nor will they be needed.
Course Materials

All of the course handouts (including lecture notes and assignments) are available in PDF format through my CS 341 homepage, whose address is given on the first page. To read the files, you will need to use a software package called Adobe Reader, which you can download for free using a link from the course web page. Be sure to check the course homepage and moodle each day since I will post announcements on it.

moodle

For this eLearning class, we will be using a computer-based group-communication system called moodle, which you can access at

http://moodle.njit.edu/

The website also provides instructions on using moodle. You must visit the moodle several times a week as I will post important announcements there in the News forum. Also, I encourage you to post general questions about the course or material there; if you don’t understand some concept or find something confusing, it’s likely that other students have the same problem, so having questions posted and answered in an open forum will benefit everyone in the class. I will respond to all questions posted on moodle.

All HW and projects must be submitted through moodle.

Lectures

As a student in an eLearning course, you will not be attending traditional “face to face” lectures but instead will watch on your computer a collection of “modules” that I recorded. The modules are videos of the lecture notes with voiceovers, and they cover the same material that I teach in a face-to-face version of CS 341. All the modules are available as video podcasts through a link posted in the course moodle webpage:

http://moodle.njit.edu/

The preface of the lecture notes describes how the recorded modules correspond to the pages in the lecture notes.

Course Policies

As a general rule, I do not give makeup exams or quizzes, I do not allow allow students to take exams or quizzes on alternate dates, nor do I allow students to turn in assignments late. Of course, if someone has a legitimate reason (e.g., jury duty, serious medical problem, conflict with a religious holiday), I will make allowances as long as you contact me beforehand (whenever it is feasible to do so) and provide proper documentation (e.g., a doctor’s note) to the Dean of Students, who will let me know if your absence can be excused. I will not accept
excuses such as having too heavy a workload, having too many exams the same week, or simply forgetting.

If upon getting back graded material (e.g., exam, project) you think that you deserve more points on a particular part, I will regrade the entire thing. Thus, you may get more points on the one part, but you may lose points on other parts. You must ask for a regrade within one week of when the graded material is returned to the class or you are informed of your score, whichever is earlier.

There are no extra-credit assignments available. So you need to make sure that you perform well on the assignments and exams.

All portable electronic devices, such as cellphones and laptops, must be turned off during exams. Students will be informed of any modifications or deviations from the syllabus throughout the course of the semester.

Homework Assignments

You must download the homework assignments from the course webpage. Except for assignment EL, do not turn in any of the homework assignments.

Assignment EL is required and will count towards 10% of your final grade. If you do not complete assignment EL by the due date given in the semester schedule below, you will lose all of the points for the assignment.

All homework solutions are posted on the web at the beginning of the semester. However, it is important that you try to work out the problems on your own without looking at the solutions first because this is the only way you will learn the material.

When working on the homework problems, be sure to show all work and give reasons (e.g., proofs) for your answers. If your proof relies on a theorem or result from the book, be sure to either state the theorem number or page number from the book. Writing out complete solutions will help you prepare for the exams, which often contain questions that are either straight from the homework or slight variations. Thus, it is important to do the homework.

Projects

The projects (programming assignments) are mandatory, and must be turned by the due date/time. Late projects will be penalized at a rate of 25 points (out of 100) per 24-hour period. For example, since Project 1 is due by 1:00pm NJ local time on 9/30/2015, you must submit all required documents for it by that date/time to not be penalized. If you turn it in after the due date/time but up to 24 hours later, then you will automatically lose 25 points of the project. If you turn in a project between 24 hours and 48 hours late, then you will automatically lose 50 points of the project. If you turn in a project between 48 hours and 72 hours late, then you will automatically lose 75 points of the project. Projects completed over 72 hours late will not be accepted.

After the first two weeks of lectures, we will have covered enough material for you to do the first project. Expect to spend at least 5-10 hours on each project, so do not wait until the last minute to try to complete it.
Academic Integrity

Students must obey the University Code on Academic Integrity, which you can access from

http://www.njit.edu/education/pdf/academic-integrity-code.pdf

Any student caught violating this will be reported immediately to the Dean of Students. Cheating includes, but is not limited to,

- communicating with others during exams
- using unauthorized materials during exams
- copying/giving a computer program from/to another person.

Semester Schedule

Unless I announce otherwise, the schedule for the semester is as below. Although you do not need to turn in the homework, you should complete the assignments according to the schedule below. The reading assignments are from the course textbook.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Modules</th>
<th>Reading</th>
<th>Homework</th>
<th>Complete By</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro, Languages</td>
<td>0, 0a to 0f</td>
<td>Chapter 0</td>
<td>HW 1</td>
<td>9/8</td>
</tr>
<tr>
<td>2</td>
<td>Regular Languages, DFA</td>
<td>0g to 1c</td>
<td>Chapter 1</td>
<td>HW 2</td>
<td>9/15</td>
</tr>
<tr>
<td>3</td>
<td>NFA, Reg Exp, Kleene's Thm</td>
<td>1d to 1i</td>
<td>Chapter 1</td>
<td>HW 3</td>
<td>9/22</td>
</tr>
<tr>
<td>4</td>
<td>Nonregular Lang, CFL</td>
<td>1j to 2a</td>
<td>Chapter 1, 2</td>
<td>HW 4</td>
<td>9/29</td>
</tr>
<tr>
<td>5</td>
<td>CFG, PDA</td>
<td>2b to 2g</td>
<td>Chapter 2</td>
<td>HW 5</td>
<td>10/6</td>
</tr>
<tr>
<td>6</td>
<td>Non-CFL, Turing Machines</td>
<td>2h to 3c</td>
<td>Chapter 2, 3</td>
<td>HW 6</td>
<td>10/13</td>
</tr>
<tr>
<td>7</td>
<td>Church-Turing Thesis</td>
<td>3d to 3i</td>
<td>Chapter 3</td>
<td>HW 7</td>
<td>10/20</td>
</tr>
<tr>
<td>8</td>
<td>Decidability</td>
<td>4a to 4e</td>
<td>Chapter 4</td>
<td>HW 8</td>
<td>10/27</td>
</tr>
<tr>
<td>9</td>
<td>Halting Problem, Reductions</td>
<td>4f to 5c</td>
<td>Chapter 4, 5</td>
<td>HW 9</td>
<td>11/3</td>
</tr>
<tr>
<td>10</td>
<td>Undecidable Problems, Big-O</td>
<td>5d to 5f, 7a, 7b</td>
<td>Chapter 5, 7</td>
<td>HW 10</td>
<td>11/10</td>
</tr>
<tr>
<td>11</td>
<td>Time Complexity, Class-P</td>
<td>7c to 7h</td>
<td>Chapter 7</td>
<td>HW 11</td>
<td>11/17</td>
</tr>
<tr>
<td>12</td>
<td>Classes P and NP</td>
<td>7i to 7m</td>
<td>Chapter 7</td>
<td>HW 12</td>
<td>11/24</td>
</tr>
<tr>
<td>13</td>
<td>Class NP-Complete</td>
<td>7n to 7r</td>
<td>Chapter 7</td>
<td>HW 13</td>
<td>12/1</td>
</tr>
<tr>
<td>14</td>
<td>Review</td>
<td>Review 1 to 6</td>
<td></td>
<td></td>
<td>12/8</td>
</tr>
</tbody>
</table>

Important Dates

The dates/times of exams and assignments are as below, subject to change.

- 9/15/2015: HW EL due
- 9/30/2015, 1:00pm NJ local time: Project 1 due
- Saturday, October 17, 2015: Midterm (tentatively 9:30am - noon)
- 10/28/2015, 1:00pm NJ local time: Project 2 due
- 11/24/2015, 1:00pm NJ local time: Project 3 due
- Saturday, December 19, 2015: Final Exam (tentatively 9:30am - noon)

1.1 Contact Information

INSTRUCTOR: Alex Gerbessiotis
OFFICE: GITC 4213, 4th floor
OFFICE HOURS: Tue 4:00-5:30pm and Thu 4:00-5:30pm
CLASS HOURS: Mon 10:00-12:55 (GITC 1205)
CLASS HOURS: By appointment Mon/Tue/Thu

1.2 Course Administration

Prerequisites: CS 280 and one of CS 241/CS 252. Last 4 digits of your NJIT id.


Course Work: 2 exams (including the final); Assignments
Grading: 1000 points = Exam1(335) + Exam2(335) + Best-5-of-7(330).
HW1-HW4 are ordinary homeworks, HW5-HW6 are programming projects, and HW7 is a paper presentation; HW5-HW7 handed out out of sequence. Each one is worth 66 points. For HW5-HW6 ONLY, a maximum of three students can work together and each one would collect the assigned graded points. HW7, the paper presentation requires a 20-minute reservation slot to be booked in advance, a one-page summary advance submission (see homework for details) and presentation.

Exams: All exams are open-textbook only. You may bring a hard-copy of the textbook but you are not allowed to borrow one during the exam or bring in class other material. Exam1 is on Mon Oct 26, 90mins. Exam2 is on Final Week, 120mins on a date to be announced by the Registrar.

Exam Conflicts: Per University regulations.

Due Dates: Paper (aka Hard-copy) submissions for HW1-HW4 before class; email submissions (txt or pdf or MSWord) by midnight the day they are due. We acknowledge email submissions promptly. It’s up to you to properly form and submit an email. Use an NJIT email address and include a Subject line as specified in Handout 0. 11 pts deducted from grade at deadline plus 2 minutes, 22 pts every 24hrs thereafter.

Tentative list of topics

T1: WebSearching: Introduction
T3: The retrieval process: Crawlers and crawling.
T4: Search Engine Architecture, Duplicate Handling
T5: Document Processing: Parsing and Tokenization
T6: Document Processing: Indexing
T7: Modeling retrieval and ranking
T8: Queries, Query processing, and Interfaces
T9: Search engine evaluation
T10: Classification and categorization
T11: Google MAPREDUCE model
T12: Case Studies: GPS
T13: Other Topics: Social Search
2.1 Course Objectives and Outcomes

Objective 1  Learn the fundamentals of Web searching.
Objective 2  Learn how a search engine works and identify the components of its architecture.
Objective 3  Learn the requirements and characteristics of web crawling, document fetching and processing.
Objective 4  Learn how to use fundamental data structures to index and store information for processing web search requests.
Objective 5  Learn the fundamentals of ranking and ranking algorithms.
Objective 6  Learn how high performance computing can benefit web searching.
Outcome 1   Be able to explain fundamental concepts related to Web searching and the architecture of search engines.
Outcome 2   Be able to identify and explain the output of search engines in the context of web searching.
Outcome 3   Be able to understand ranking and indexing algorithms and their limitations.
Outcome 4   Be able to design a search engine architecture based on input design requirements.
Outcome 5   Be able to effectively use high performance computing in the design of a Web search infrastructure.
Outcome 6   Be able to effectively apply ranking algorithms.

2.2 Tentative Course Calendar

<table>
<thead>
<tr>
<th>Week</th>
<th>Mon</th>
<th>HWOut</th>
<th>HWin</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>Tue* 9/8</td>
<td>HW5 out</td>
<td>HW6 out</td>
<td>HW5, HW6 are mini-projects</td>
</tr>
<tr>
<td>W2</td>
<td>9/14</td>
<td>HW1 out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W3</td>
<td>9/21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W4</td>
<td>9/28</td>
<td>HW2 out</td>
<td>HW1in</td>
<td></td>
</tr>
<tr>
<td>W5</td>
<td>10/5</td>
<td>HW3 out</td>
<td>HW2in</td>
<td></td>
</tr>
<tr>
<td>W6</td>
<td>10/12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W7</td>
<td>10/19</td>
<td></td>
<td>HW3in</td>
<td></td>
</tr>
<tr>
<td>W8</td>
<td>10/26</td>
<td>Exam1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W9</td>
<td>11/02</td>
<td>HW4 out</td>
<td></td>
<td>Mon Nov 2: Withdrawal Deadline</td>
</tr>
<tr>
<td>W10</td>
<td>11/09</td>
<td>HW5 in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W11</td>
<td>11/16</td>
<td>HW4 in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W12</td>
<td>11/23</td>
<td></td>
<td></td>
<td>Thanksgiving week:Tue is a Thu</td>
</tr>
<tr>
<td>W13</td>
<td>11/30</td>
<td>HW7?</td>
<td></td>
<td>HW7 presentation?</td>
</tr>
<tr>
<td>W14</td>
<td>12/07</td>
<td>HW6in, HW7</td>
<td>HW7 presentation</td>
<td></td>
</tr>
<tr>
<td>W15</td>
<td></td>
<td>Exam2**</td>
<td>Tue Dec 15- Mon Dec 21</td>
<td>is Final Exam Week</td>
</tr>
</tbody>
</table>

* First day of classes is the Tuesday after Labor Day (9/8) that is "Monday" for NJIT  ** Check with the Registrar

Any modifications or deviations from these dates, will be done in consultation with the attending students and will be posted on the course Web-page. It is imperative that students check the Course Web-page regularly and frequently.
Grading  Written work will be graded for conciseness and correctness. Be brief and to the point and write clearly. Programming problems will be graded based on test instances decided by the instructor on an AFS machine (afsconnect1, afsconnect2, or osl11). Do not expect partial credit if your code fails to run on all test instances, and you do not provide a bug report.

Grades  Check the marks in written work and report errors promptly. Resolve any issue no later than the Reading Day. For students who submit programming work or have a paper presentation, an email with your grade will be sent back to you. The final grade is decided based on a 0 to 1000 point performance. A 50% or more is C or better, 85-90% or more usually guarantees an A.

Collaboration  Collaboration of any kind is NOT allowed in the in-class exams and the homeworks. An exception to this rule is HW5-HW6 that explicitly allow collaboration (teams of no more than 3); in such a case collaboration is allowed between members of the team only for the specific homework only. Students who turn in work/answers to questions sourced through the Internet or otherwise, or is product of another person/student’s work, risk severe punishment, as outlined by the University. The work you submit must be the result of your own effort.

Mobile Devices  Mobile phones/devices and/or laptops/notebooks MUST BE SWITCHED OFF (NOT JUST SILENCED) before the class exams. Switch off noisy devices before class.

Email/SPAM  Send email from an NJIT email address. NJIT spam filters or we will filter other email address origins. Use the appropriate subject line as specified in Handout 0. Include cs345 in the subject line then.

Missing class  If you miss a class and there is no Exam or Homework due it’s up to you to make up for lost time.

Missing Exam  If you miss an exam and there is a valid documentation for your absence, such documentation must be presented within 3 working days from the day the reason for the absence is lifted. The maximum accommodation will be the number of missing days to the exam date. You also need to present your case to the Dean of Student Services (DOSS); we will respond after receiving confirmation from DOSS.

Missing HW  If you are sick (see Missing Exam for the procedure) there is no notion of a make-up homework or delayed submission of a homework other than the penalties specified on page one of this document. Per DOSS and Instructor approvals, a homework grade might get extrapolated from the final exam grade (EX2).

Programs  Follow submission guidelines for HW5-HW6, if you plan to do it/then.

Presentation  Follow submission guidelines for HW7, if you plan to do it.

The NJIT Honor Code will be upheld; any violations will be brought to the immediate attention of the Dean of Students. Read this handout carefully!
CS 356 Computer Networking

Instructor: George Blank
Office: GITC 4404
NJIT Phone: (973) 596-5485 Home Office (973) 625-0803
Office Hours: Monday 3:00-4:30 pm; Tuesday 1-2:15 and Friday 10:30-11:45 am.

Course Prerequisite: None, but persons with no previous programming experience usually have difficulty competing with classmates for high grades.

Course Description: This course provides an introduction to computer networks, with a special focus on the Internet architecture and protocols. Besides the theoretical foundations, students also acquire practical experience by programming reduced versions of real Internet protocols.

Core Topics

A. Protocol layering and the Internet architecture.
B. Application layer protocols: DNS, HTTP, Email.
C. The client-server model and the Socket interface.
D. IP addressing, subnets, CIDR.
E. Data link layer/LAN overview, ARP, RARP/BOOTP/DHCP.
F. IP protocol, ICMP.
G. Routing: link state routing and distance vector routing. OSPF, RIP, BGP.
H. UDP, TCP (includes flow control, congestion control, RED).
I. NAT, VPN, IPv6.

Additional Topics

1. More application layer protocols: FTP, Telnet, SSH
2. Recent topics: Peer-to-peer networks, sensor networks, mobile networks
3. IP multicast.
4. QoS, Integrated and differentiated services.

Textbook
Assignments

Programming Assignment:

Three Part Semester Long Programming Project

1) Basic socket application to exchange information over a network. Ideally, the server should be multi-threaded.

2) Link state or distance vector routing emulation: Two processes emulate routers that exchange topology information to build their routing tables.

3) Extend part 2 to Four separate processes operating across a network. The user should be able to add/remove links and add/remove routers.

Homework Assignments: Eight sets of problems and/or review exercises derived from the textbook. Because the solutions manual for the text is widely available on the Web, data in the problems will be changed. The solutions must be discussed during the first lecture after the graded homework was returned.

Wireshark Labs: Eight Lab assignments done on a network protocol analyzer

Wireshark Labs:

Final grades are limited by class rank. Only the top 20% of the class are eligible for a grade of A, and only those in the next 20% are eligible for a grade of B+. Note that a particular average such as 90% does not guarantee a particular grade. Also, A students are expected to complete all assignments on time, and a grade of B+ requires substantially all assignments completed on time.

Professor Blank subscribes to the NJIT Honor Code. Persons submitting the work of others as their own or cheating on the exams may receive a grade of XF (Failed due to cheating.) Most of the homework assignments use code from the text. You must use this code, but you must also identify it. It is important when submitting assignments to include comments in your code to identify the source of any code you use that is not original. Ideally, your original work should be in a different color or easily identifiable.

Student Outcomes

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
   1. Students will be able to calculate delays and achievable TCP throughput in the Internet.
2. Students will be able to calculate routing paths for two main classes of routing protocols: link-state and distance-vector.

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
1. Students will be able to analyze the trade-offs for building a computer network in terms of throughput, delay, reliably and security, and be able to recommend different approaches to meet different needs.

(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
1. Students will be able to create computer programs that pass information across a network using the socket interface.
2. Students will be able to create multithreaded server programs.

(i) An ability to use current techniques, skills, and tools necessary for computing practice.

Metrics Used to measure student outcomes listed above:

For (a) 1.
Homework Problems: 1) Propagation Delay & Transmission Delay, Processing Delay
2) Improvements from Cache 3) Client/Server vs. Peer-to-Peer throughput
Exam Questions
For (a) 2.
Homework Problems: 5) Dijkstra’s shortest path algorithm
6) Bellman-Ford distance vector algorithm
Exam Questions
For (b) 1.
Homework Problems: 4) checksum calculation 6) Reverse Path Forwarding
7) Cyclic Redundancy Check Calculation 8) CSMA throughput
Network Labs: HTTP, DNS, UDP, TCP, IP, Ethernet, ARP, and SSL
Exam Questions
For (c) 1 and 2:
A semester long programming project developing a distance vector router simulator that communicates across a network using sockets, uses routing algorithms to maintain and update router tables, and simulates four routers running separately on different hosts
For (i).
Developing programs on distributed networks
Labs using network analyzer
Class Exercises using tools such as ping, ifconfig, ipconfig, netstat, traceroute, dig
Exam Questions
Fundamentals of Network Security

Course No.
CS 357

Sections
101

Title
Fundamentals of Network Security

Course Websites
http://www.cs.njit.edu/~karvelas/CS-357-F15

Prerequisite(s)
CS-356 or some other undergraduate course in the area of networking

Instructor
Dennis Karvelas
Office Room No.: GITC 4212
Office Phone: 973-596-2987
Fax: 973-596-5777
Email: dionissios.karvelas@njit.edu
Website: http://www.cs.njit.edu/~karvelas

Instructor Office Hours
Mondays: 5.10 pm - 5.50 pm
Tuesdays: 4.30 pm - 5.50 pm
Thursdays: 5.10 pm - 5.50 pm

Description
This course offers an in depth study of network security issues, types of computer and network attacks, and effective countermeasures. It provides both a theoretical foundation in the area of security and hands-on experience with various attack tools, firewalls, and intrusion detection systems. Upon successful completion of the course, students will have gained a deep understanding of: i) the wide variety of threats facing organizations with on-line presence, ii) the techniques used by attackers to find and exploit the vulnerabilities of computer systems and networks, iii) how to detect and design effective defenses against those attacks.

Grading Scheme
Two Quiz: 10% each
Two lab assignments: 10% each
Midterm: 28%
Final: 32%
Topics
Introduction to Network Security
Internetworking Protocols
Target Reconnaissance
Target Scanning and Vulnerability Analysis
Gaining Access Techniques
Maintaining Access Techniques
Techniques for Hiding Attacker's Files, Processes, and Network Traffic.
Denial of Service Attacks (DoS) and Distributed DoS (DDoS)
Designing an Effective Defense
Firewalls and Intrusion Detection Systems
Incident Response

Text Book(s)
Additional material will be posted on the class website for some of the topics of the course

Time & Place
CS-357-101: Wednesdays, 10.0 AM, KUPF-211

Other Info
Honor Code:
The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students. Note in particular that copying programming assignments or exam papers, in full or in part is forbidden.
Fall 2015 CS433 Intro Linux Kernel Programming - Course Syllabus

Check the announcements regularly.

- Homework submission page: http://moodle.njit.edu
- Instructor: Andrew Sohn, GITC 4209, (973)596-2315, email: sohn at cs dot njit dot edu
- Office Hours: Tue 11:30am-1pm, Thur 5:30-7pm
- Teaching Assistant: No TA - NJIT policy but my Ph.D. student Yang Li yl98_at_njit_dot_edu who was TA for this course for several years who is working in the City is willing to help you and answer any questions you may have for the course.
- Class time and location: TR 4:00pm-5:25pm KUPF 203; See the registrar's page http://www.njit.edu/registrar/schedules/courses/fall/2015F.CS.html
- Recommended books:
  - Computer Organization and Design, Morgan Kaufmann, Patterson and Hennessy (strongly recommended; read and understand before you graduate)
- Grading: 8 to 10 kernel programming assignments (20%), Test 1 (20%), Test 2 (20%), final exam (40%)
- Exam dates: Test 1, Thur, 10/1/2015; Test 2, Thur, 10/29/2015; Final exam date and time - see the registrar's page.
- See Academic Integrity

Schedule by Week - Contents may change according to class pace

1. LAMP, virtualization, data center computing infrastructure, Review of Intel architecture based on Intel 64 and IA-32 Architectures Software Developer's Manual (3429 pages!), Intel and AT&T Linux assembly, assembly in-line programming
2. Setting up LXR (Linux cross referencer) on your laptop and in-class demonstration, compiling the kernel, Module programming, Booting - machine BIOS, disk MBR, Grub Linux loader, preliminary setup (setup(), startup_32() 1 and 2)
3. Booting continues, Overview of kernel startup and initialization (start_kernel)
4. Memory - overview, segmentation, mapping, paging (get_free_pages),
5. Memory - paging (get_free_pages), caching (kmalloc)
6. Memory - caching (kmalloc) and process address space (vmalloc)
7. Process - thread union, task and thread struct, PID0 (swapper), PID1 (init),
8. Process - PID2 (kthread), PID3 (softirqd), PID4 (migration), process scheduling
9. Process - process switching
10. Interrupts - exceptions (traps), hard interrupts do_IRQ()
11. Interrupts - soft interrupts do_softirq(), ksoftirqd, timer interrupts
12. File system - short intro to file system vfs_read(): virtual file system, registering, mounting, block IO, IO scheduler, device driver, read/write
13. Networking - short intro to sending packets sys_send(): User, BSD, Inet, TCP, IP, Softirq, ISR, NIC.
CS 490: Design in Software Engineering

Syllabus

Instructor

Instructor: Theodore L. Nicholson
Office: GITC 4414
Office Hours: Tuesday 4:00 - 5:00 pm, Friday 4:00 - 5:00 pm
Email: theodore.l.nicholson@njit.edu

Course Description

This course focuses on the methodology for developing software systems. Methods and techniques for functional requirements analysis and specifications, design, coding, testing and proving, integration and maintenance are discussed.

Textbook

Software Engineering (9th Edition)
Ian Sommerville
ISBN: 978-0137035151

Grading Scheme

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exam (week 11)</td>
<td>30%</td>
</tr>
<tr>
<td>Course Project</td>
<td>70%</td>
</tr>
<tr>
<td>--Alpha</td>
<td>(5%)</td>
</tr>
<tr>
<td>--Beta</td>
<td>(15%)</td>
</tr>
<tr>
<td>--Release Candidate</td>
<td>(20%)</td>
</tr>
<tr>
<td>--Final Version</td>
<td>(30%)</td>
</tr>
</tbody>
</table>

Student Outcomes

- Students will be able to explain the major theories and methods applicable to professional software engineering.
- Students will be able to design, implement and evaluate a computer based system to meet desired needs.
- Students will be able to function effectively on a team to accomplish a goal.
- Students will be able to use current techniques, skills and tools necessary for computing practice.

Topics
- Software Processes
- Project Management
- Software Requirements/Requirements Engineering Process
- System Models
- Architectural Design
- Distributed System Architectures
- Application Architectures
- Object-Oriented Design
- User Interface Design
- Tools – debuggers, unit testing, profiling, version control.

**Cheating Policy**

Cheating on a programming assignment results in zero credit for all students involved. Cheating on an exam will result in an "F" in the course.

You may discuss problems with each other, in fact, you are encouraged to do so. Where does discussion end and cheating start? You may **NOT** copy lines of code from anybody or anywhere. You may **NOT** use code in your assignments that you did not write. You may not use third party frameworks. As a general rule: If you don't understand the code and can't explain the code, you can't use the code.

Please familiarize yourself with the **NJIT Honor Code**. Violations of the honor code will be dealt with seriously and reported immediately to the Dean of Students.

**Late Policy**

To receive credit all programming assignments must be handed in on time. No credit will be given for any programming assignment that is not turned in on the day (and time) it is due.

**Prerequisites**

CS 280, CS 288
Syllabus

Faculty Contact Information

Instructor: Jonathan Kapleau
Email: kapleau@njit.edu
Office Hours: Tuesdays 3:00-4:00 PM ET

Course Description

Intensive introduction to computer science principles with emphasis on methodology and problem solving. Topics include program design, introductory data structures, and algorithms and their analysis. Programming language is fully discussed and serves as the vehicle for the concepts.

Textbook and Materials


Course Outcomes
Module 07: Inheritance and Polymorphism
Module 08: Exception Handling
Module 09: Recursion
Module 10: Introduction to Data Structures
Module 11: Stacks
Module 12: Queues
Module 13: Generalized Linked Structures
Module 14: Trees
Course Surveys

a. Gain familiarity and comfort with the basic functionality.
b. Represent data using the appropriate data type.
c. Evaluate a situation in order to choose the appropriate data type.
d. Evaluate a situation in order to choose the appropriate data type.
e. Use pre-existing and programmer defined classes.
f. Process homogeneous data.
g. Determine when to use inheritance to facilitate software design.
h. Determine when to use exception handling to handle errors.
i. Determine when to use recursion instead of iteration.
j. Learn how to properly use encapsulation.
k. Use stacks.
l. Use queues.
m. Use generalized linked structures.

Grading Categories

<table>
<thead>
<tr>
<th>Categories</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm Exam</td>
<td>30%</td>
</tr>
<tr>
<td>Projects</td>
<td>35%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>35%</td>
</tr>
</tbody>
</table>

Course Structure

Module Number   | Dates                   |
----------------|-------------------------|
1               | September 1 – Sep       |
2               | September 8 – Sept      |
3               | September 15 – Sep      |
4               | September 22 – Sep      |
5               | September 29 – Oct      |
Late Work and Make-Up Exams

To receive full credit, all programming assignments must be submitted on time. Assignments that are not submitted on time will be penalized based on the following late policy:

- The type and severity of the penalty will be determined by the instructor.
- In general, a deduction of 10 points (out of 100) for the first day late, 30 additional points for the second day, 30 additional points for the third day, and 100 points for the fourth day late will be subtracted from the final grade of the assignment.

Sharing Information

Cheating on a programming assignment results in zero credit for the assignment. Programming assignments may NOT be solved in collaboration with other students. Cheating on an exam will result in a grade of 0 for the exam.

You may discuss problems with each other. Where does discussion start? You may NOT copy lines of code from anybody or any source other than the code specifically stated in the assignment. Cheating on an exam will result in a grade of 0 for the exam.

Please familiarize yourself with the NJIT Honor Code. Violation of the Honor Code will be dealt with seriously and reported immediately to the Dean.

Student Conduct

The NJIT University code on academic integrity will be followed.

Student with Disabilities Codes
NJIT adheres to section 504 of the Rehabilitation Act (ADA) and accommodations are provided at no cost to the student. If you would like additional information, please contact Dr. Phyllis Brim-Counseling and Psychological Services (C-CAPS), Campbell 205, (973) 596-3420. For further information, visit the Student website.

Technical Support

For assistance with the following items, please contact NJIT at: 1-973-596-2900 or http://nist.njit.edu/support/helpdesk.php

- UCID
- Library database access
- Webmail by Google email system
- Sessions
- Password assistance*

For assistance with the following items, please contact Technical Support at: 1-888-789-0407 or https://supportcenter.embanet.com/NJIT

- Content within this course
- Assignments
- Discussion Forums
- Quizzes
- All other items related to the running of this course

*NJIT passwords may be changed using the Global Password tool. You will need to know your current UCID and UCID password referred to 1-973-596-2900.

Periodic changing of passwords and strategies for managing passwords are encouraged for anyone using a computer. All members of the university community are encouraged to review tips for password management and to check the information regularly.

Software and Hardware Requirements

Sometimes, you will be required to use Word processing and spreadsheet software such as MS Word and PowerPoint found in Microsoft Office.
comfortable with various aspects of using the Internet such as:

- Search engines
- Newsgroups
- Email
- Ability to download files

To view certain media elements in this course, you will need to plug-ins such as Shockwave, Flash, and Adobe Acrobat on your links in the course to download and install the appropriate software. **Important:** With regards to plug-ins, ensure you are using the version of each plug-in you require. View the hardware and software requirements for each course.

**Browser Check**

To test your web browser for compatibility in the Moodle environment, visit the Online Support Center page and select the 'Check My Browser' link on the right side.

---

**Student Services and Support**

Heather Minton  
<h.minton@onlineprograms.njit.edu>  
877-615-8696, press 2

Schedule a phone appointment with Heather: https://www.vciforum.

Contact the program coordinator for:

- Change of address, phone, or email
- If you have not received your course materials
- If you are having difficulty contacting a faculty member
- If you have difficulty completing your course work due
- To drop/withdraw from a course
- General program information
Weekly Listing of Course Topics

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6
Review. First Midterm.

Week 7
Existential Quantifier. Universal Quantifier.

Week 8

Week 9

Week 10

Week 11
Review. Second Midterm.
Week 12
Analysis of Recursive Algorithms.

Week 13
Introduction to Graphs.

Week 14
Introduction to Trees.
CS 602 - Syllabus

Course Title: Advanced Java and World Wide Web Programming (CS 602)

Course Prerequisite: Placement Test

Course Description: Advanced Web-based programming with an emphasis on the Java language and platform. No prior knowledge of Java is required but students are expected to have a good understanding of Object Oriented Programming concepts such as Encapsulation, Inheritance and Polymorphism, and experience with the C++ programming language (See the course prerequisite, above).

This graduate level course commences with the basic constructs and syntax of the language but quickly moves to the core advanced features which include such topics as Networking and Sockets, Remote Method Invocation (RMI), Database Connectivity (JDBC), Java Beans, Multi-Threading and Lightweight Components (Swing).

Text:

Object Oriented Software Development Using Java (2nd Edition) by Xiaoping Jia
Addison Wesley; 2nd edition (November 1, 2002)
ISBN-10: 0201737337

Instructor: Theodore L. Nicholson

Office: 4414 GITC
Office Hours: Tuesdays & Fridays 4:00 - 5:00 pm
E-mail: theo@njit.edu (= best way to contact)

Web Site: http://web.njit.edu/~theo/

Course Topics:

Topic IA Introduction to Java: Overview

Topic IB Comparison to C++

Topic 2 Basic Language Features: Primitives, Objects, Constructors, Variables, Methods, Classes, Access Specification

Topic 3 Basic Language Features Continued: Inheritance; Essential Java Classes

Topic 4 Operators; Sequence, Selection and Repetition; Exception Handling; Inner Classes

Topic 5 Interfaces; Event Handling; Layout Managers

Topic 6 Abstract Window Toolkit (AWT) Event Handling; Streams

Topic 7 Streams; Swing
Topic 8 Multithreading

Topic 9 Networking

Topic 10 Remote Method Invocation (RMI)

Topic 11 Java Database Connectivity (JDBC)

Topic 12 Java Beans; Applets v. Applications

Grades: Midterm Exam (30%), Final Exam (35%), Final Project (35%)
CS 602 Advanced Java and the Web

Course Title: Advanced Java and World Wide Web Programming

Instructor: George Blank, GITC Room 4404 gblank@njit.edu.
NJIT Phone: (973) 596-5485 Home Office (973) 625-0803
Office Hours: Monday 3:00-4:30 pm; Tuesday 1-2:15 and Friday 10:30-11:45am.
Web Site: http://web.njit.edu/~gblank/
This course uses http://moodle.njit.edu/ as the primary communication platform.

Course Prerequisite: None, but graduate courses in the Department of Computer Science presume that the student has an undergraduate degree in Computer Science or a similar field. The department has a Bridge Program for students who do not have the necessary background. Students have registered for this course without the necessary background, and some have completed it successfully, but most have difficulty and many end up dropping the course.

Course Description: Advanced Web-based programming with an emphasis on the Java language and platform. No prior knowledge of Java is required but students are expected to have a good understanding of Object Oriented Programming concepts such as Encapsulation, Inheritance and Polymorphism.

This graduate level course commences with the basic constructs and syntax of the language but quickly moves to the core advanced features which include such topics as Networking and Sockets, Remote Method Invocation (RMI), Database Connectivity (JDBC), Java Beans, Multi-Threading and Lightweight Components (Swing).

Weekly homework assignments including programs written in Java are required for the first two thirds of the course. The last third of the course requires developing an individual student project. There are two exams, both requiring the development of a Java program in a limited amount of time.

Text:

Topic 1 Object-Oriented Software Development
Topic 2 Object-Oriented Modeling Using UML
Topic 3 Introduction to Java
Topic 4 Elements of Java
Topic 5 Classes and Inheritance
Topic 6 From Building Blocks to Projects
Topic 7 Design by Abstraction
Topic 8 Object-Oriented Application Frameworks
Topic 9 Design Case Study
Topic 10 Design Patterns
Topic 11 Concurrent Programming
Topic 12 Distributed Computing
Topic 13 Security
These topics generally follow the chapters in the textbook. Some materials are presented out of sequence to allow more time for developing code and to adjust to different amounts of information in each chapter. There are also supplementary materials from the instructors personal experience as a corporate software developer and software development manager.

Grades: First Exam (25%), Second Exam (25%), Final Project (25%), Web Site Homework (25%)
Final grades are determined by class rank. The top 25% of the class gets a grade of A, the next 25% get a grade of B+. Note that a particular average such as 90% does not guarantee a particular grade. Also, persons who submit work late more than once and classroom students who miss two or more software inspections are not eligible for grades higher than B.

Professor Blank subscribes to the NJIT Honor Code. Persons submitting the work of others as their own or cheating on the exams may receive a grade of XF (Failed due to cheating.) Most of the homework assignments use code from the text. You must use this code, but you must also identify it. It is important when submitting assignments to include comments in your code to identify the source of any code you use that is not original. Ideally, your original work should be in a different color or easily identifiable.
Prerequisites

CS 114 (Intro. to Computer Science II), or CS505 or equivalent; Math 226 (Discrete Math).

Objective

This course is an introduction to the study of computer algorithms, with the goal of developing the ability to construct efficient algorithms.

Learning Outcomes

- The ability to design programs using advanced data structures and algorithms.
- The ability to analyze the resources used by an algorithm.

Textbook


You may find the following references useful:


Grading

Several homework problems will be assigned over the semester. You may discuss the problems in general terms with your colleagues, but you must come up with your own solutions. No late submissions will be accepted. Any programs written as part of the homework must be written in Java or C++.

There will be an in-class midterm exam on Oct. 15. There will be a final exam, at a time and place to be determined by the registrar. The grade will be based on: final 40%, midterm 30%, homework 20%, and class participation 10%. In addition, a passing grade on the final is required to pass the course. It is not possible to raise a grade by doing additional work after the end of the semester.
If you are unable to meet any of the course requirements (for example due to illness), you must contact me immediately (leave a phone message or email) and supply documentation for an excuse.

**Academic Honesty**

It is every student's responsibility to understand and adhere to the provisions of the academic honor code. You may discuss homework problems with your colleagues, but all written work must be your own. Copying programs or written assignments from any source is a serious violation of the academic honor code. Any evidence of dishonesty will be reported to the Dean of Students for disciplinary action.

**Tentative Course Outline**

2. Search trees, skip lists.
3. Design techniques: divide and conquer, greedy method, dynamic programming.
5. Graph algorithms: traversals, shortest paths, minimum spanning trees.
Syllabus

Faculty Contact Information

Instructor: Ali Mili
Email: mili@njit.edu
Office Hours: TBD

Contact Technical Support’s 24/7 helpdesk at 1-888-789-0407 if you have any issues accessing the course content, discussion forums, quizzes, course email, and/or submitting/posting assignments.

Course Description

This is a graduate-level course on data-structures and algorithms, with an emphasis on algorithm design techniques and analysis of algorithms. Introductory topics include analysis techniques, worst-case and average-case analysis, induction, recursion, recurrence relations, and divide-and-conquer design technique. Advanced topics include priority queues, hash tables, binary-search trees, balanced search trees (AVL trees), sorting algorithms, as well as other design techniques such as greedy-method and dynamic-programming, graph algorithms, and text processing algorithms.

Textbook and Materials


Course Outcomes

- Students will be able to assess and analyze algorithms.
- Students will be able to implement various problem solving techniques.
- Data modeling / data representation.
- Fundamental algorithms (optimization).
Assessments

There are seven assessments throughout the course: three text assignments (6 points each), two programming assignments (12 points each), a midterm exam (15 points), and a final exam (25 points).

Refer to the Schedules and Deadlines page for information regarding due dates, and the individual course pages for assignment and exam details.

Grading Scale

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentile</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.0</td>
<td>90-100%</td>
</tr>
<tr>
<td>B+</td>
<td>3.5</td>
<td>85-89.99%</td>
</tr>
<tr>
<td>B</td>
<td>3.0</td>
<td>80-84.99%</td>
</tr>
<tr>
<td>C+</td>
<td>2.5</td>
<td>70-79.99%</td>
</tr>
<tr>
<td>C</td>
<td>2.0</td>
<td>60-69.99%</td>
</tr>
<tr>
<td>F</td>
<td>N/A</td>
<td>&lt;= 59.99%</td>
</tr>
</tbody>
</table>

Please note: This table is given as an approximate guidance; the instructor reserves the right to depart from it significantly, depending on the grade distribution of the class.

Course Structure

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Start and End Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>September 1st—September 7</td>
</tr>
<tr>
<td>2</td>
<td>September 7—September 13</td>
</tr>
<tr>
<td>3</td>
<td>September 14—September 20</td>
</tr>
<tr>
<td>4</td>
<td>September 21—September 27</td>
</tr>
<tr>
<td>5</td>
<td>September 28—October 4</td>
</tr>
<tr>
<td>6</td>
<td>October 5—October 11</td>
</tr>
<tr>
<td>7</td>
<td>October 12—October 18</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>October 19—October 25</td>
</tr>
<tr>
<td>8</td>
<td>October 26—November 1st</td>
</tr>
<tr>
<td>9</td>
<td>November 2nd—November 8</td>
</tr>
<tr>
<td>10</td>
<td>November 9—November 15</td>
</tr>
<tr>
<td>11</td>
<td>November 16—November 22</td>
</tr>
<tr>
<td>Thanksgiving</td>
<td>November 23—November 29</td>
</tr>
<tr>
<td>12</td>
<td>November 30—December 6</td>
</tr>
<tr>
<td>12</td>
<td>December 7—December 10</td>
</tr>
<tr>
<td>Final Exam</td>
<td>December 18</td>
</tr>
</tbody>
</table>

Time Commitment

The students are expected to allocate ten hours per week to study and work on the assignments for this course.
Course Policies

All correspondence with the instructor will take place through Moodle. Questions of general interest should be posted on Moodle’s Ask the Instructor discussion board. Personal/individual questions should be emailed if the instructor is not responsive to your discussion board posting. If you do so, please start your subject header with the sequence "CS610: [Spring, Summer, or Fall]," and then the year.

Submission Deadlines

All assignments are due by 11:55 p.m. ET on the dates designated in the Schedule and Deadlines page.

Programming Assignments

Programming Assignments must be completed in either JAVA or a C variant (C, C++, C#). Java freeware can be downloaded at http://java.com/en/download/manual.jsp for Linux, Mac OS X, Solaris and Windows based machines. For those using C variant languages on Windows, Microsoft Visual Studio 2012 Professional can be downloaded through NJIT’s IT department (http://ist.njit.edu/software/download.php). Those using C variant languages on other operating systems should VPN into NJIT’s Public Computing Site using their UCID to use required programs.

Students must include useful documentation within their program. Programs must run on either a Linux, UNIX, or Windows based machine. Source code will be uploaded to Moodle by the due date. The uploaded assignment should contain the following:

- Assignment Upload:
  - Student’s name
  - Course and section number
  - Assignment number
  - Any special instructions/comments about running your program
  - Problems: If your program is not fully running, indicate to what extent it runs and where it has problems.
  - Source code of your program
  - Input(s)
  - Output produced by your program

- Submissions for programming assignments must include three components:
  - the source code, preferably in pdf format.
  - three or four sample executions of the program, showing the cycle of user interactions (submitting inputs, posting outputs).
  - the program in some executable form, or a URL where I can execute your program remotely. Please do not assume that I have a copy of your compiler, or that I can download it. I also cannot recreate your run-time environment.

Late Work and Make-Up Exams
No late homework will be accepted. In case of missing an exam, a make-up may be taken only after providing written documentation to the Dean of Students. The Dean of Students will then inform the instructor about their decision. **Please be advised that no homework or programming assignment will be accepted or graded unless it is submitted on time and through Moodle.**

---

**Sharing Information**

Students are free to discuss assignments with their colleagues. However, they should not take any written (electronic or otherwise) record away from the discussion. This applies when the assignment is supposed to be an individual effort or whenever two teams discuss common problems they are each encountering (inter-group collaboration). After the discussion, it is advisable to engage in at least half hour of non-course related activity before starting to work on the assignment. This will assure that students are able to reconstruct by themselves what they learned from the discussion.

---

**Student Conduct**

The NJIT University code on academic integrity, found at http://www.njit.edu/doss/policies/conductcode/article4.php, will be followed in all courses.

---

**Student with Disabilities Codes**

Appropriate accommodations are provided at no cost to the student. If you have any questions or would like additional information, please contact Dr. Phyllis Bolling, Center for Counseling and Psychological Services (C-CAPS), Campbell Hall, (entry level), room 205, (973) 596-3420. For further information, visit the Student Disability Services website.

---

**Technical Support**

For assistance with the following items, please contact NJIT IST Helpdesk at: 1-973-596-2900 or http://ist.njit.edu/support/helpdesk.php

- UCID
- Library database access
- Webmail by Google email system
- Sessions
- Password assistance*

For assistance with the following items, please contact Technical Support at: 1-888-788-0407 or https://supportcenter.embanet.com/NJIT

- Content within this course
- Assignments
- Discussion Forums
- Quizzes
- Exams
- All other items related to the running of this course

*NJIT passwords may be changed using the Global Password Change mechanism. You will need to know your current UCID and UCID password. Questions can be referred to (973) 596-2900.

Periodic changing of passwords and strategies for managing them are best practice for
Software and Hardware Requirements

Sometimes, you will be required to use Word processing and presentation software, such as MS Word and PowerPoint found in Microsoft Office. You will also need to be comfortable with various aspects of using the Internet such as:

- Search engines
- Newsgroups
- E-mail
- Ability to download files

To view certain media elements in this course, you will need to have several browser plug-ins such as Shockwave, Flash, and Adobe Acrobat on your computer. Use the links in the course to download and install the appropriate software application.

Important: With regards to plug-ins, ensure you are using the most recent version of each plug-in you require. View the hardware and software requirements for this course.

Browser Check

To test your web browser for compatibility in the Moodle environment, go to the Online Support Center page and select the 'Check My Browser' link located on the right side.

Student Services and Support

Heather Minton
h.minton@onlineprograms.njit.edu
877-615-8696, press 2

Schedule a phone appointment with Heather: https://www.vcita.com/v/hminton

Contact the program coordinator for:

- Change of address, phone, or email
- If you have not received your course materials
- If you are having difficulty contacting a faculty member
- If you have difficulty completing your course work due to personal issues
- To drop/withdraw from a course
- General program information

| Last modified: Monday, August 10, 2015, 4:58 PM |

2015 Fall - CS 610851

© Copyright 2015 Moodlerooms Inc. All Rights Reserved
Department of Computer Science

CS 631: Data Management Systems Design
Fall Semester 2015-Sessions 001 and 002 - Dimitri Theodoratos

Objective

The objective of the course is to give a thorough introduction to the concepts of designing, organizing, querying and managing relational databases. It will be balanced between theory and practice.

The course covers nine topics: Conceptual Modeling (the Entity-Relationship model and the Enhanced Entity-Relationship model), the Relational Model (Concepts, Integrity Constraints, Update Operations, Relational Algebra, Relational Calculus), From Conceptual Models to Logical Models (Mapping the ER Model to the Relational model), the Commercial Database Language SQL (Data Definition, Basic Integrity Constraints, Schema Changes, Basic and Complex Queries, Aggregate Functions and Grouping, Data Change Statements, Views, Complex Constraints, Embedded SQL, Cursors, Dynamic SQL, JDBC), Active Databases (Active Database Rules, Triggers), Database Physical Organization (File Organizations, External Hashing, Indexing, B+ trees), Query Evaluation (Operator Evaluation, Algorithms for Relational Operators, Query Optimization), Database Design Theory (Functional and Other Dependencies, Dependency Inference, Normal Forms, Schema Decomposition, Normalization), and Database Management Systems Implementation Issues (Transaction Processing, Concurrency Control).

You will learn the latest of the SQL standard and you will get hands on experience on creating, populating, querying, maintaining and managing a relational database using the commercial database management system Oracle.

Course organization

- The slides for each lecture are available before the class. A good practice is to read from your book the material to be taught in class and to come prepared.
- After the theory on a certain topic is presented, you can download the corresponding questions and exercises which you should try to solve. Most of them are discussed in class in subsequent lectures. Three homework assignments will contain exercises on selected topics. Their solutions will be posted on the web page of the class in due time.
- Alongside your assigned homework you should work in groups on a project which has three deliverables during the semester.
- Project deliverables are to be handed in the day they are due or earlier. A project demonstration is required for the third deliverable of the project.
- There will be two exams: a midterm halfway through the semester, and a final on the last day of the class.
- Lecture slides, exercises, homework assignments, project requirements, and other material (e.g. Entity-Relationship diagrams and solutions of homework assignments) will be available for downloading in due time on this web page. Important announcements will be also placed on it. Bookmark the course web page. It is a "living document" and you have to visit it at least once a week.

Time and Place

CS631-001:  Date and time: Tuesday, 10.00AM - 12.55PM
Room: KUPF 204

CS631-103:  Date and time: Tuesday, 09.00PM - 09.15PM
Room: KUPF 117

Personal

Instructor:
Dimitri Theodoratos
Office: GITC 4208
Office hours: Wednesday: 4.00 - 5.30 PM.
Thursdays: 3.00 - 4.30 PM.
Office hours are valid during exam periods, holidays, and breaks
If you cannot meet my office hours, send me an email to schedule an appointment.

Web page: http://web.njit.edu/~dth
Email: dth@njit.edu

Teaching assistant:
Ananya Dasgupta
Office: GITC 4201
Email: adasgupta@njit.edu

Prerequisites
Knowledge of a programming language, algorithms, data structures, file organization and the basics of the relational data model is required.

Textbook


Other books (not required)

**GENERAL TEXTBOOKS**


**DATABASE THEORY**


**DBMS IMPLEMENTATION**


**SQL**


**Tentative lecture schedule**

<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture #</th>
<th>Chapters in Textbook</th>
<th>Content</th>
<th>Project deliverables and assignments due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, Sep. 1</td>
<td>1</td>
<td>1, 2, 3</td>
<td>Introduction, Database Concepts, Database Systems Architecture, Data Models, Conceptual Modeling.</td>
<td></td>
</tr>
<tr>
<td>Tuesday, Sep. 2</td>
<td>2</td>
<td>3, 4</td>
<td>No Class - Monday classes meet</td>
<td></td>
</tr>
<tr>
<td>Tuesday, Sep. 15</td>
<td>3</td>
<td>5</td>
<td>The Entity-Relationship and the Extended Entity-Relationship Model</td>
<td></td>
</tr>
<tr>
<td>Tuesday, Sep. 22</td>
<td>4</td>
<td>8</td>
<td>The Relational Model - Basic Definitions, Integrity Constraints, Update Operations, Formal query languages - Relational Algebra.</td>
<td></td>
</tr>
<tr>
<td>Tuesday, Sep. 22</td>
<td>4</td>
<td>8</td>
<td>Formal query languages - Relational Algebra.</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Topic</td>
<td>Assignment</td>
<td>Due Date</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Sep. 29</td>
<td>5</td>
<td>Mapping Entity-Relationship Schemas to Relational Schemas.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oct. 6</td>
<td>6</td>
<td>Commercial Query Languages. SQL -- Data Definition and Simple Queries.</td>
<td>Assignment 1</td>
<td>Oct. 14</td>
</tr>
<tr>
<td>Oct. 13</td>
<td>7</td>
<td>SQL -- More Complex Queries, Updates, Views, Constraints</td>
<td></td>
<td>Oct. 21</td>
</tr>
<tr>
<td>Oct. 20</td>
<td>8</td>
<td>Embedding SQL into a host language.</td>
<td>Project deliverable 2</td>
<td>Oct. 21</td>
</tr>
<tr>
<td>Oct. 27</td>
<td>9</td>
<td>Midterm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nov. 3</td>
<td>10</td>
<td>Active Databases and Triggers.</td>
<td>Assignment 2</td>
<td>Nov. 25</td>
</tr>
<tr>
<td>Nov. 10</td>
<td>11</td>
<td>Functional and Other Dependencies, Normal Forms, Schema Decomposition, Normalization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nov. 17</td>
<td>12</td>
<td>Primary File Organizations, External Hashing, Indexing</td>
<td>Assignment 3</td>
<td>Dec. 9</td>
</tr>
<tr>
<td>Nov. 24</td>
<td>13</td>
<td>Query Evaluation -- Operator Evaluation, Algorithms for Relational Operators, Query Optimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec. 1</td>
<td>14</td>
<td>Transaction Processing, Concurrency Control.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec. 8</td>
<td>15</td>
<td>Final</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Attendance and Participation**

You are supposed to attend all the classes. Participation is highly encouraged to make the class more interactive. Class attendance and participation are taken into consideration by the instructor for the evaluation of the students. Experience shows that students that do not attend the classes do not perform well in the midterm and final exams. If you miss one class be sure to consult one of your classmates about the content of the lecture and visit the course web page to get notes, exercises, assignments, deadlines and announcements.

**Homework Assignments**

There will be three homework assignments. You should work on the homework assignments individually. Solutions are provided for you after the due date to compare with your own solutions. If you work reasonably on them you will get all the points.

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 1</td>
<td>The Relational model, Relational Algebra.</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>Queries and Constraints in SQL and Active Databases.</td>
</tr>
<tr>
<td>Assignment 3</td>
<td>Formal Database Design Theory.</td>
</tr>
</tbody>
</table>

The homework assignments can be downloaded here:

Assignment 1. Relational Model, Relational Algebra
Assignment 2. Queries and Constraints in SQL and Active Databases.

**Project**

A project to design and implement a database using a DBMS will be assigned. It will proceed progressively through conceptual design, logical design, populating the database, and implementing application programs. The methodology for database development learned in class should be used.

**Project groups**

Groups of at most 3 students are required. You can choose the classmates you want to work with. You can also work by yourselves if you prefer. In order to form a group you have to fill a form available in class or in my office during office hours. You do not have to express a preference. If you do not express any preference, you will be put in a group randomly.

**Project phases**

...
The project has three phases that cover the areas described below. Each phase has a different weight in the course grade. A report (deliverable) for each phase has to be handed in by the due date shown above in the lecture schedule table. For every phase of the project, one deliverable per group needs to be handed in (not one for each member of the group). Each group needs to contact the TA to schedule an appointment for demonstrating its project at the end of the semester (dates will be made available).

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1</td>
<td>Analysis &amp; Conceptual Design</td>
</tr>
<tr>
<td>Phase 2</td>
<td>Logical Design</td>
</tr>
<tr>
<td>Phase 3</td>
<td>Implementation and Testing</td>
</tr>
</tbody>
</table>

**Project Resources**

For the project you have to use a DBMS. You can use Oracle or MySQL at NJIT. You can also download and install to your PC a DBMS. Several are available for free download.

Students registered for CS 631 can get MySQl and Oracle accounts at NJIT. University Computing Systems (UCS) has streamlined the database accounts provisioning process. Students can now create either (or both) a MySQL or Oracle account simply by running the database password reset utility at http://mypassword.njit.edu/db.

The utility will send the database owner an email describing the login credentials for either a new account or the existing account. The email is sent to the owner's official UCID@njit.edu address and should arrive within an hour (generally much sooner).

The procedure requires that the database owner have an NJIT UCID login and they know the password. Support will be provided by the IST Help Desk at http://help.njit.edu.

Database accounts will no longer expire. Access to their database is granted to all NJIT students until they leave NJIT.

**Oracle:**
The database server prophet.njit.edu in NJIT supports coursework on Oracle.

Prophet uses internal Oracle authentication. This means that a user will need to log onto Oracle with an Oracle userid and password. The userid will generally be the user's UCID, but the password will NOT be the user's AFS password.

Users will _NOT_ be able to log onto prophet.njit.edu. They will need to access Oracle via clients.

SQL*Plus is a program of Oracle that provides an interactive environment with the Oracle database server. You can type commands directly at the SQL> prompt or have SQL*Plus execute commands stored in files. Sql*plus clients are available on all solaris AFS clients. To access your Oracle account on prophet enter /usrsql/-bannerplus on the command line.

A quick Oracle guide including SQL*Plus (by Ramakrishnan and Gehlke).

**Oracle resources and services (registration is free) from Oracle Technology Network.**

**Underground Oracle FAQ.**

**Aqua Data Studio:**

To connect remotely to Oracle or MySQL at NJIT you can use Aqua Data Studio, Aqua Data Studio is a database developer's complete Integrated Development Environment. Info about Aqua Data Studio at http://list.njit.edu/support/db/prophet.php#ADS.

Aqua Data Studio 6 is available at NJIT but it is maybe easier to install Aqua Data Studio 6.5.8.

To install Aqua Data Studio 6.5.8, go to http://list.njit.edu/software/display.php?d=283. In order to download it, you should either be on campus or have a VPN connection (for VPN access, http://vpn64.njit.edu).

Install the program following the instructions at http://list.njit.edu/software/documentation/aqua_data_studio_650.php. As explained in the last step of installation instructions, you will need a license key. You can obtain this key from http://list.njit.edu/software/display.php?d=283 by clicking at the "Request product key" button.

After a successful installation, you need to register the oracle server at NJIT in your program, so that you do not have to enter server information every time you want to connect. Information for registering a server can be found at http://list.njit.edu/software/documentation/aqua_data_studio_register_server.php for both MySQL and Oracle. Once registered the servers will appear on the left panel of the program.

**Homework Submission and Late policy**

Assignments and project deliverables should be submitted on or before the day and time they are due through Moodle.

The 3rd and final project deliverable should be handed in the time of the project demonstration.

Late submissions will not be accepted or will get penalties.
Grading

The midterm, the project and the final exam contribute to the course grade as follows:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>30%</td>
</tr>
<tr>
<td>Homework Assignments</td>
<td>5%</td>
</tr>
<tr>
<td>Project</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>5% each of the 1st and 2nd deliverables, and 15% the 3rd deliverable</td>
</tr>
<tr>
<td>Final</td>
<td>40%</td>
</tr>
</tbody>
</table>

Notes/Slides

<table>
<thead>
<tr>
<th>Notes/Slides:</th>
<th>Exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BASIC CONCEPTS</strong></td>
<td></td>
</tr>
<tr>
<td>Databases and Database Users</td>
<td></td>
</tr>
<tr>
<td>Database Systems Concepts and Architecture</td>
<td></td>
</tr>
<tr>
<td><strong>CONCEPTUAL MODELING</strong></td>
<td></td>
</tr>
<tr>
<td>Conceptual Modeling, the ER model and the Enhanced ER model</td>
<td></td>
</tr>
<tr>
<td><strong>THE RELATIONAL MODEL: CONCEPTS AND FORMAL QUERY LANGUAGES</strong></td>
<td></td>
</tr>
<tr>
<td>Basic Definitions, Integrity Constraints, Update Operations</td>
<td></td>
</tr>
<tr>
<td>Relational Algebra</td>
<td></td>
</tr>
<tr>
<td>Relational Algebra Example Queries</td>
<td></td>
</tr>
<tr>
<td><strong>FROM CONCEPTUAL MODELS TO LOGICAL MODELS</strong></td>
<td></td>
</tr>
<tr>
<td>Mapping the ER and EER Models to the Relational Model</td>
<td></td>
</tr>
<tr>
<td><strong>SQL: A COMMERICAL DATABASE LANGUAGE</strong></td>
<td></td>
</tr>
<tr>
<td>Data Definition, Basic Integrity Constraints, Schema Changes</td>
<td></td>
</tr>
<tr>
<td>Basic Queries</td>
<td></td>
</tr>
<tr>
<td>Complex Queries, Aggregate Functions and Grouping, Advanced Querying Issues</td>
<td></td>
</tr>
<tr>
<td>Data Change Statements, Views and Complex Constraints</td>
<td></td>
</tr>
<tr>
<td>Using SQL in an Application</td>
<td></td>
</tr>
<tr>
<td><strong>TRIGGERS</strong></td>
<td></td>
</tr>
<tr>
<td>Active Database Rules and Triggers</td>
<td></td>
</tr>
<tr>
<td><strong>FORMAL DATABASE DESIGN THEORY</strong></td>
<td></td>
</tr>
<tr>
<td>Functional dependencies (FDs), FD Inference, FDs and Keys</td>
<td></td>
</tr>
<tr>
<td>Normal forms</td>
<td></td>
</tr>
<tr>
<td>Schema decomposition, Normalization, Other Dependencies</td>
<td></td>
</tr>
<tr>
<td><strong>DATABASE PHYSICAL ORGANIZATION</strong></td>
<td></td>
</tr>
<tr>
<td>File Organizations</td>
<td></td>
</tr>
<tr>
<td>External Hashing</td>
<td></td>
</tr>
<tr>
<td>Indexing</td>
<td></td>
</tr>
<tr>
<td><strong>DATABASE MANAGEMENT SYSTEMS IMPLEMENTATION ISSUES</strong></td>
<td></td>
</tr>
</tbody>
</table>
Midterm and Final Exam

The midterm and final will be open book exams. You can also have with you my slides/notes. Nothing else is allowed. The exams will be composed of several exercises/questions to be answered.

Academic Integrity

Violations of the academic integrity in this class will not be tolerated. Cheating in assignments or exams will result in an "F" grade for the student. Read the Academic Honor Code of Njit. The Njit Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students.

Links

ACM SIGMOD
IEEE TCOD
The VLDB Endowment
The RDB Endowment
DBLP - Database and Logic Programming Bibliography
Interactive Online SQL Training
Advanced Online SQL Training
# Advanced Database Systems Design

<table>
<thead>
<tr>
<th>Course No.</th>
<th>CS 632</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sections</td>
<td>001</td>
</tr>
<tr>
<td>Title</td>
<td>Advanced Database Systems Design</td>
</tr>
<tr>
<td>Course Website</td>
<td><a href="http://web.njit.edu/~geller/632/Protected/index.html">http://web.njit.edu/~geller/632/Protected/index.html</a></td>
</tr>
<tr>
<td></td>
<td>This page is password protected.</td>
</tr>
<tr>
<td>Prerequisite(s)</td>
<td>CS 631 AND Good knowledge of programming in any modern high-level procedural programming language such as Java or C++.</td>
</tr>
<tr>
<td>Instructor</td>
<td>James Geller</td>
</tr>
<tr>
<td></td>
<td>• Office Room No.: 4104 GITC Building</td>
</tr>
<tr>
<td></td>
<td>• Office Phone: 973 596-3383 (I am more likely to reply to email than to phone!!!)</td>
</tr>
<tr>
<td></td>
<td>• Fax : 973-596-5777 (Nobody is using fax anymore, really.)</td>
</tr>
<tr>
<td></td>
<td>• Email : <a href="mailto:james.geller@njit.edu">james.geller@njit.edu</a></td>
</tr>
</tbody>
</table>
| Instructor Office Hours | FR 1:00 – 2:30  
TH 4:00 – 5:30  
FRI 11:30-2:30 BY APPOINTMENT ONLY  
(Subject to FREQUENT change. There are always meetings I have to go to. Email me before you come.) |
| Description | This course has four parts.  
1) Review your SQL and learn more of it.  
2) Learn PL/SQL.  
3) Study small parts of other languages that are needed in system building. We are interested in systems that have a Web frontend and a relational database backend. For this purpose we will use PSP (PL/SQL Server Pages).  
4) Study a few advanced topics on databases, such as XML, OODBs, DB security, parallel and distributed databases, No-SQL databases and performance tuning.  
The purpose of the course is to get better in programming in an Oracle environment, especially in PL/SQL and to learn about building systems that have a relational database as a backend and the Web as a frontend. |
<table>
<thead>
<tr>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PL/SQL as &quot;normal&quot; language</td>
</tr>
<tr>
<td>• PL/SQL accessing tables</td>
</tr>
<tr>
<td>• Using AQUA for Oracle programming</td>
</tr>
<tr>
<td>• Object-Oriented Modeling</td>
</tr>
<tr>
<td>• Inserting etc. in PL/SQL</td>
</tr>
<tr>
<td>• Object-Oriented DBs</td>
</tr>
<tr>
<td>• Triggers in PL/SQL</td>
</tr>
<tr>
<td>• Object-Relational DBs</td>
</tr>
<tr>
<td>• Objects in SQL</td>
</tr>
<tr>
<td>• HTML (brief)</td>
</tr>
<tr>
<td>• Intro to XML</td>
</tr>
<tr>
<td>• Database Security</td>
</tr>
<tr>
<td>• XML in Oracle SQL</td>
</tr>
<tr>
<td>• Querying XML</td>
</tr>
<tr>
<td>• Distributed Databases</td>
</tr>
<tr>
<td>• PSP Programming</td>
</tr>
<tr>
<td>• Introduction to UNIX/LINUX use (brief)</td>
</tr>
<tr>
<td>• Web databases</td>
</tr>
<tr>
<td>• No-SQL databases</td>
</tr>
<tr>
<td>• Intro to HBase</td>
</tr>
<tr>
<td>• Indexing and Performance Tuning</td>
</tr>
</tbody>
</table>

Topics are subject to change or omission, depending on time. The catalog description of the class will be updated in the near future.

<table>
<thead>
<tr>
<th>Text Book(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) REQUIRED:</td>
</tr>
<tr>
<td>Oracle 10g Programming: A Primer (Paperback)</td>
</tr>
<tr>
<td>by Rajshekhar Sunderraman</td>
</tr>
<tr>
<td>Paperback: 438 pages</td>
</tr>
<tr>
<td>Publisher: Addison Wesley;</td>
</tr>
<tr>
<td>Copyright: 2008</td>
</tr>
</tbody>
</table>

If there are not enough copies in the NJIT bookstore, then please get it at www.amazon.com or www.barnesandnoble.com.

<table>
<thead>
<tr>
<th>Text Book(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2) RECOMMENDED:</td>
</tr>
<tr>
<td>Fundamentals of Object-Oriented Design in UML</td>
</tr>
<tr>
<td>Meilir Page-Jones</td>
</tr>
<tr>
<td>Publisher: Addison Wesley</td>
</tr>
<tr>
<td>Copyright: 2000</td>
</tr>
</tbody>
</table>
(This book was NOT ordered in the NJIT bookstore.)

I personally recommend to students (that can afford it...) to buy books. Thus having books on Java, HTML, XML and JavaScript is a good idea, even if we will not use them a lot. Or use the Web. :)

3) **WEB MATERIAL AND HANDOUTS:**
There will be handouts in class and reading materials as well as videos that can be found on the Web.

| Time & Place | THURSDAY 10:00AM – 12:55AM;  
|             | Place: CKB = Central King Building, Room 222. |
| Other Info  | Exams:  
|             | There will be two Midterm Exams, and one Final Exam.  
|             | There will be a prerequisite quiz.  
|             | The prerequisite quiz will not count towards your final grade.  
|             | However, if you fail the prerequisite quiz I will ask you strongly to drop the class (to allow somebody else into the class who is waiting for a spot).  
|             | Homework:  
|             | There is a prerequisite makeup. (You don’t need to do it if you get full credit on the prerequisite quiz.)  
|             | There will be four homework assignments consisting mostly of PL/SQL programming. However, there will be additional questions and problems. Late policies will be announced together with the assignments. In general, there will be late penalties. |
|             | Grading:  
|             | The assignments will be assigned points from the following maximum: |
|             | Prerequisite Quiz: 0  
|             | Midterm 1: 100  
|             | Midterm 2: 100  
|             | Final: 200  
|             | Prerequisite Makeup: 10  
|             | Homework 1: 40  
|             | Homework 2: 50  
|             | Homework 3: 50  
|             | Homework 4: 50  
|             | Total 600 |
At the end of the semester, I will add up your total points and curve the results for the whole class.

The department has voted on a stringent new curve for all courses. Only the top 25% of the class should expect an A.

Furthermore, up to 20% of the class may face a C+, C, or F. In my experience, students getting 550 out of 600 points tend to be in the top 25%, but this is no guarantee for the future.

Also note that most students typically get all the points on the Homework assignments. Thus your position in the curve and your class grade depend almost entirely on the exams. On the other hand, missing a single homework is highly likely to lower your grade at least one letter grade.

**Computer Use:**
You have to get an AFS account (ID), if you don't have one.

You will need a printer, Web access and a text editor (notepad will do).

Media Services can be reached at (973) 596-3005.

**Oracle and Aqua:**
You will use the Oracle database system and PL/SQL. PL/SQL is part of the normal Oracle distribution.

You will use *Aqua Data Studio* for as interface to Oracle. Here is information on Aqua.

You may download Aqua Data Studio VERSION 15 from this URL: [http://ist.njit.edu/software/download.php](http://ist.njit.edu/software/download.php)

**Aqua Activation:**

Please note that access to this commercial software is highly restricted. You need to follow the ACTIVATION steps exactly, or it won't work. *Exactly.*

[http://ist.njit.edu/software/ads/12/Activate.php](http://ist.njit.edu/software/ads/12/Activate.php)

Originally Aqua is from: [http://www.aquafold.com/downloads.html](http://www.aquafold.com/downloads.html)
You may find documentation for Aqua Data Studio from this URL:  
https://www.aquaclusters.com/app/home/project/public/aquadatastudio/wiki
book/Documentation11/page/0/Aqua-Data-Studio-11-0

This is for version 11 and you are downloading version 12 right now.  
The labs have an older version of Aqua installed.  
However, as we don’t use sophisticated features, the versions are  
almost equivalent for us.

**NJIT Passwords:**

If there is an NJIT password problem, try this:  
https://mypassword.njit.edu/  
This is the UCID password reset.

You need a DIFFERENT password for Oracle.  
How do you get that different password?

Go to:

https://mypassword.njit.edu/db

In order to work from home on Oracle you need to install VPN.  
See Here: http://telecom.njit.edu/vpn/

The computer we are using is called **prophet.njit.edu**  
The connection identifier that you will be asked for is **course**.

Furthermore, you can get extensive explanations of the ORACLE setup by  
going to: http://ist.njit.edu/support/db/oracle.php

We will use a SECOND Oracle account for Oracle-Web  
programming. Details of this account will be announced in class. Don’t  
worry about it now.

**Cheating:**

The NJIT Honor Code will be upheld.

Any violation will be brought to the immediate attention of the Dean of  
Students.
Cheating on an assignment or exam will of course result in 0 credit.

You may "talk" about Homework assignments with each other. Where does talking end and cheating start? You may NOT copy lines of code from anybody. You may NOT use code in your program where you don't understand WHY it works, even if it works, and even if you wrote it yourself.

A few years ago I caught a student who offered my homework on a commercial (bidding) Web site. In other words, he paid money to have somebody else do my assignments for him. I will be checking for this, and I will have no mercy if I catch you. The student in question was failed out of the course. Who knows, the person bidding to do your homework might be me.

In addition, I will give questions out of the homework assignments on the midterm and final exam. If your answers on the exams are substantially wrong, even though you did the homework correctly I might question you about the homework and reduce your homework credit to zero, from whatever it was before.
CS 634 Data Mining

Instructor
Name: Dr. Jason Wang
Office: GITC 4211
Phone: (973) 596-3396
Email: wangj@njit.edu

Prerequisite
Knowledge of a programming language (e.g., C, C++, Java, Python, Perl, R, SQL, MATLAB, etc.) is required.

Course Description
Covers the concepts and principles of association rule mining, decision trees, clustering, Web information retrieval and integration, Web mining, time series data mining and graph mining. Hands-on experiences include the design and implementation of (1) an association rule mining tool for transactional databases, (2) a data clustering tool (or a text mining tool), and (3) two data classification tools. Some of these projects can be implemented in a cloud computing environment such as Amazon web services. Students can also choose projects on Big Data mining with Hadoop/MapReduce, or either CouchDB or MongoDB technologies.

Course Objectives
To familiarize students with basic data mining principles, modern data mining methods and tools, as well as advanced data mining applications, and to help students find jobs in fields related to data mining, data science, data analytics, data management and Big Data. Specifically, the student will be able to

- Explain data mining concepts, principles and methods,
- Use a wide range of publicly available data mining tools,
- Evaluate the effectiveness and efficiency of these data mining tools based on different performance measures,
- Design, develop and implement customized data mining algorithms, heuristics, methods, techniques and software tools.

Recommended Course Textbooks (these books are optional, not required)


Course Outline and Schedule
This course covers the concepts and techniques of

1. Association Rule Mining

2. Classification, Regression and Prediction
3. Clustering
4. Text Mining
5. Web Searching, Mining and Crawling
6. Mediators, Wrappers and Data Warehousing
7. Time Series Data Mining
8. Graph Mining
9. Advanced Data Mining
10. New Applications

**Course Workload**
There will be one midterm project, one final term project, one term paper, and one exam.

**Course Grade**
Midterm project -- 25%, Final term project -- 30%, Term paper -- 10%, Exam -- 35%.

**Grading Scale**
A: 93% and above; B+: 86%-92.9%; B: 78%-85.9%; C+: 70%-77.9%; C: 60%-69.9%; F: Below 60%.

**Honor and Policy**

- Students found cheating or plagiarizing will be immediately referred to the Dean of Students and the NJIT Committee on Professional Conduct and subject to Disciplinary Probation, a permanent marking on the record, possible dismissal, and an "F" grade in the course. All submitted assignments will be checked for similarities, and plagiarism and guilty students identified.
- In the exam, each student is required to sign the Honor Code Agreement "On my honor, I pledge that I have not violated the provision of the NJIT Student Honor Code."
CS 639 Elec Medical Records: Comp Imp
(Medical Informatics)  
Dr. Karen Hare

<table>
<thead>
<tr>
<th>Syllabus</th>
<th>Fall 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course No.</td>
<td>CS 639</td>
</tr>
<tr>
<td>Sections</td>
<td>101</td>
</tr>
<tr>
<td>Title</td>
<td>Medical Terminologies and Electronic Medical Records (In the catalog abbreviated as Elec Medical Records: Comp Imp)</td>
</tr>
<tr>
<td>Day/Time</td>
<td>Wednesday, 6:00pm to 9:05pm</td>
</tr>
<tr>
<td>Location</td>
<td>Tiernan Hall, Room 106</td>
</tr>
<tr>
<td>Prerequisite(s)</td>
<td>CS 631 or Undergraduate Database course or practical experience with a Database system or Permission of Instructor. Some knowledge of programming in any modern high-level procedural programming language such as Java or C++ is helpful but not required.</td>
</tr>
<tr>
<td>Instructor</td>
<td>Dr. Karen Hare</td>
</tr>
<tr>
<td>GT/IC CS Dept. 4th Floor, mailbox</td>
<td></td>
</tr>
<tr>
<td>Dept. phone number: 973-596-3366</td>
<td></td>
</tr>
<tr>
<td>Dept. fax number: 973-596-5777</td>
<td></td>
</tr>
<tr>
<td>Email: <a href="mailto:kxh1868@njit.edu">kxh1868@njit.edu</a></td>
<td></td>
</tr>
<tr>
<td>Instructor Office Hours</td>
<td>Location: TBD</td>
</tr>
<tr>
<td>By Appointment</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>This course presents an introduction to Medical Informatics for Computer Science students by covering important topics from Medical Informatics. We will be evaluating the design, use and auditing of medical terminologies and; principles of Electronic Medical Records, and its closely related “brethren” Electronic Health Records and Personal Health Records.</td>
</tr>
<tr>
<td></td>
<td>Designed to introduce you to the organization, structure, and operation of the nation’s health care system in such a way to help you identify more effectively your present and future roles as consumer, provider, manager, decision maker, and analyst. For over 50 years, the United States has faced a health care crisis. We have struggled with the fundamental goals of providing cost-effective, high-quality care to all Americans. Yet politically, economically, and socially acceptable answers have continued to be debated and unresolved. As a health care manager, these questions will be the background to your future roles and this class.</td>
</tr>
<tr>
<td></td>
<td>Introductions to Medical Informatics assume a good knowledge of the language of medicine, which may not be assumed for Computer Science students. However, Medical Informatics makes heavy use of computational tools and databases and is an area of intensive commercial growth. Thus, students taking this course will be prepared to work in what is undoubtedly a growth industry, namely medical software development.</td>
</tr>
<tr>
<td></td>
<td>This course is intended for graduate Computer Science students. Graduate Information Systems students can benefit from this course. We hope to attract some Medical students and Information Technology professionals who might be</td>
</tr>
</tbody>
</table>
CS 639 Elec Medical Records: Comp Imp  
(Medical Informatics)  

Dr. Karen Hare  
kxh1868@njit.edu  

Syllabus  
interested in a deeper understanding of computational and information technology issues in the healthcare environment.

This is not a programming course. However, there will be hands-on work with several browsers and EMR systems. The class will be held in a hybrid format with mixed lecture/seminar format and virtual online session (TBD). The Professor will lecture roughly half of the class time. After that we will read and discuss chapters in the text book or papers that will be handed out and/or do hands-on experiments with EMR systems and terminology systems. Every student will be expected to contribute to discussions.

**BRING YOUR LAPTOP if you have one.**

**This syllabus is a work in progress. It may change over the course of the term. Changes made by instructor supersede this syllabus.**

<table>
<thead>
<tr>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction and Historical Overview</td>
</tr>
<tr>
<td>Electronic Health Records/Electronic Medical Records/Personal Health Records</td>
</tr>
<tr>
<td>Federal Legislation</td>
</tr>
<tr>
<td>HIPAA, Security and Privacy issues</td>
</tr>
<tr>
<td>SNOMED, MEDCIN System (Hands on experiments)</td>
</tr>
<tr>
<td>ICD-9, ICD-10</td>
</tr>
<tr>
<td>Overview of UMLS (Unified Medical Language System)</td>
</tr>
<tr>
<td>Meaningful Use of EHR/EMR systems</td>
</tr>
<tr>
<td>ONC, HITECH,CMS, IOM, NIH</td>
</tr>
<tr>
<td>Health Information Exchange, Interoperability Issues, Infrastructure Integration</td>
</tr>
<tr>
<td>Problem lists</td>
</tr>
<tr>
<td>Future of EHR/EMR systems; Commercial and Open Source systems</td>
</tr>
</tbody>
</table>

**Topics are subject to change or be omitted, depending upon time.**

<table>
<thead>
<tr>
<th>Text Book(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REQUIRED</strong></td>
</tr>
</tbody>
</table>
The above numbers include “MyHealthProfessionsKit” Access Card students need to get in and download the software for the course. |
Publisher: Springer  
ISBN: 978-1-4471-4473-1 |

**NJIT Bookstore should have both books**

<table>
<thead>
<tr>
<th>Time &amp; Place</th>
</tr>
</thead>
</table>
| 6:00pm to 9:05pm  
Tiernan Hall, Room 106 |
### Syllabus

**Exams:**
There will be one Midterm Exam, and one Final Exam. All exams are face-to-face.

**Homework:**
There will be 5 homework assignments involving working with EMR systems and terminology browsers. Reading, article reviews, writing or Web research will be required. All assignments must be in a Word processing format.
*** There will be a penalty of 10 point per day for late assignments unless otherwise arranged.
*** There will be a penalty for not being prepared for class discussion.
*** Students with special needs are to contact the instructor by the end of week 1.

**Attendance and Participation:**
Repeated absence is STRONGLY discouraged and will logically and definitely lead to students not doing well on exams. A portion of your grade will be based on participation and quizzes.

**Grading:**
The assignments will be assigned points from the following maximum:
- Midterm: 200
- Final: 200
- Homework 1: 25
- Homework 2: 50
- Homework 3: 25
- Homework 4: 50
- Homework 5: 50

Total 600

At the end of the semester, I will add up your total points and curve the results for the whole class. Normal curving rules of the CS Department will be applied. Note, that most students typically get all the points on the Homework assignments. Thus, your position in the curve and your class grade depend almost entirely on the exams. On the other hand, missing a single homework is highly likely to lower your grade at least one letter grade.

<table>
<thead>
<tr>
<th>University Resources</th>
<th>Plagiarism and Academic Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The approved “University Code on Academic Integrity” is currently in effect for all courses. Should a student fail a course due to a violation of academic integrity, they will be assigned the grade of “XF” rather than the “F” and this designation will remain permanently on their transcript. All students are encouraged to look over the University Code on Academic Integrity <a href="http://www.njit.edu/education/pdf/academic-integrity-code.pdf">here</a> and understand</td>
</tr>
</tbody>
</table>
Syllabus

this document. Students are expected to uphold the integrity of this institution by reporting any violation of academic integrity to the Office of the Dean of Students http://www.njit.edu/doss/. The identity of the student filing the report will be kept anonymous. NJIT will continue to educate top tier students that are academically sound and are self-disciplined to uphold expected standards of professional integrity. Academic dishonesty will not be tolerated at this institution.

Student Disability Services
NJIT is committed to providing students with documented disabilities equal access to programs and activities. If you have, or believe that you may have, a physical, medical, psychological, or learning disability that may require accommodations, please contact Student Disability Services. Information on the self-identification, documentation and accommodation process can be found on the webpage at: http://www.njit.edu/counseling/services/disabilities.php

Getting Help - General
The IST Helpdesk is the central hub for all information related to computing technologies at NJIT. This includes being the first point of contact for those with computing questions or problems.

There are three ways to contact the Helpdesk:
1. Call 973-596-2900. Monday - Friday 8 am - 7 pm.
2. Go to Student Mall Room 48. Monday - Friday 8 am - 7 pm
3. Log a Help Desk Service Request online - https://ist.njit.edu/support/contactus.php

Getting Help – Moodle
In addition to the Helpdesk, NJIT has a number of resources available to help you learn/use Moodle. Please be aware of the following:
1. Getting Started Using Moodle (Student Course):
http://njit.mrooms.net/course/view.php?id=6204
2. Student Moodle Tutorials: http://moodle.njit.edu/tutorials/students/index.php
3. Student Moodle FAQs: http://moodle.njit.edu/tutorials/students/faq.php

Working From Home
1. In order to work from home on certain systems you need to install VPN. Site: http://telecom.njit.edu/vpn/
CS 643 101: Cloud Computing, Fall '15

Wednesdays, 6-9:05pm, CKB 314

Instructor

Cristian Borcea
Office: GITC 4410
Phone: 973 596-2866
Office Hours: Mondays 5pm-7pm and by appointment

borcea@njit.edu

News

- 08/29 The weekly topics and readings as well as the papers to be presented in class are posted below. I also posted the slides for the first lecture. I emailed you the credentials to access these materials.

Short description

The course presents a top-down view of cloud computing, from applications and administration to programming and infrastructure. Its main focus is on parallel programming techniques for cloud computing and large scale distributed systems which form the cloud infrastructure. The topics include: overview of cloud computing, cloud systems, parallel processing in the cloud, distributed storage systems, virtualization, security in the cloud, and multicore operating systems. Students will study state-of-the-art solutions for cloud computing developed by Google, Amazon, Microsoft, Yahoo, VMWare, etc. Students will also apply what they learn in one programming assignments and one project executed over Amazon Web Services.

Learning outcomes

Upon the successful completion of this course, the student should be able to:

- Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure.
- Compare the advantages and disadvantages of various cloud computing platforms.
- Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.
- Program data intensive parallel applications in the cloud.
- Analyze the performance, scalability, and availability of the underlying cloud technologies and software.
- Identify security and privacy issues in cloud computing.
- Explain recent research results in cloud computing and identify their pros and cons.
- Solve a real-world problem using cloud computing through group collaboration.

Why take this course?

Cloud computing represents a major paradigm shift in computing from the era of personal computers to the era of computing as utility. Most major Internet services are already deployed in the "the cloud." In the near future, we may store all our data in "the cloud" and execute most applications from "the cloud." This course is aimed at all graduate students (both M.S. and Ph.D. students) who want to learn how to design and program cloud services as well as how to build and administer cloud systems. By studying real-world systems developed in industry during the past few years, students will acquire cutting-edge knowledge that may be a major advantage when searching for a job.

Prerequisites

CS 656 or CS 633 or instructor's permission. If you didn't take CS 656 or CS 633, but you would like to take this class, you should come and talk with me about your background. Basic Unix/Linux skills and good programming skills are necessary for the assignment and the project.

Lectures and Readings

There is no book required for this class. Each lecture is based on recent papers/articles covering a specific topic. Students are required to read the papers before the class and participate in the discussions in class.

Paper Presentations

Students will present, in groups of three or two, one research paper during the semester. These papers cover very recent developments in cloud computing. The presentations (using power point slides) will take place in class, and extra-credit will be assigned for active participation in discussions.

Programming Assignment

There will be one individual programming assignment consisting of creating an AMI for Hadoop and implementing short Hadoop programs on the Amazon Web Services platform.

Project
Students will choose their project topic and work in teams of four or three to design, implement, and evaluate cloud applications using Hadoop on the Amazon Web Services platform. General requirements for all projects will be discussed in class after the midterm.

**Exams**

There will be two exams: a midterm, and a final exam. Both exams are closed book (i.e., papers, notes). The final exam will cover only the material taught after the midterm. In case of missing an exam, a make-up may be taken only after providing written documentation to the Dean of Students.

**Homework**

Homework will be assigned 4 times during the semester to prepare students with the type of questions encountered in exams. This homework is not for credit, but students are encouraged to do it. The solutions will be discussed in class.

**Grading**

- Midterm exam - 25%
- Final exam - 25%
- Programming Assignment - 20%
- Project - 20%
- Paper presentation - 10%

**Schedule**

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Required readings</th>
<th>Papers to be presented in class</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Cloud Platforms I. Amazon Web Services, Google AppEngine.</td>
<td>• Amazon Web Services Documentation.</td>
<td>•</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Google AppEngine Documentation.</td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Course Topics</td>
<td>Additional Resources</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td></td>
</tr>
</tbody>
</table>
- Advices on reading and presenting research papers |
- Hadoop Documentation |
- M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, *Dryad: Distributed Data-Parallel Programs from* |

- E. Boutin et al., *Apollo: Scalable and Coordinated Scheduling for*
<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Readings</th>
<th>Notes</th>
</tr>
</thead>
</table>
| 10   | Virtualization I. VMWare virtual machine monitor. Discussion of homework 3 solutions. | - J. Sugerman, G. Venkitachalam, and B-H. Lin, Virtualizing I/O Devices on VMWare Workstation's Hosted Virtual Machine Monitor, Usenix 2001. | - J. Gonzalez et al., GraphX: Graph Processing in a Distributed Dataflow Framework, Usenix OSDI 2014.  
- S. Muralidhar et al., |
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Text</th>
</tr>
</thead>
</table>
- D. Yuan et al., *Simple Testing Can Prevent Most Critical Failures. An Analysis of Production Failures in Distributed Data-
Academic Integrity

The University Code of Academic Integrity will be upheld, and any violations will be brought to the immediate attention of the Dean of Students.

Modifications to Syllabus

The students will be consulted and must agree to any modifications or deviations from the syllabus throughout the course of the semester.
CS 643-103: Cloud Computing

Fall 2015

Monday: 6:00 ~ 9:05PM CKB214

Instructor: Xiaoning Ding
Phone: (973)596-3390
E-mail: xiaoning.ding@njit.edu
Office Location: GITC 4203
Office Hours: Monday: 4:00am ~ 6:00pm, Thursday: 4:00 pm ~ 6:00pm, and by appointment

Course Description
This course presents a top-down view of cloud computing, from applications and administration to programming and infrastructure. Its main focus is on parallel programming techniques for cloud computing and large scale distributed systems which form the cloud infrastructure. The topics include: overview of cloud computing, cloud systems, parallel processing in the cloud, distributed storage systems, virtualization, secure distributed computing, and multicore programming.

Prerequisites
CS 633 Distributed Systems or CS 656 Internet and Higher-Layer Protocols.

Lecture Material
The course does not have a textbook. The instructor will introduce some background (access Moodle for lecture notes). Then, students will be asked to read papers and technical reports in the related areas. Students will also be asked to present and discuss the materials in the classes.

Course Work:
- Midterm exam - 25%
  A midterm will be scheduled in the 8th week of the semester. Students are not allowed to refer to course materials (papers, technical reports, or lecture notes).
- Final exam - 25%
  Final exam is not cumulative. The exam will be scheduled by the University. Check online for the time and location.
- Programming Assignment - 20%
  There will be one individual programming assignment consisting of creating an AMI for Hadoop and implementing short Hadoop programs on the Amazon Web Services platform.
- Project - 20%
  Students will choose their project topic and work in teams of four or three to design, implement, and evaluate cloud applications using Hadoop on the Amazon Web Services platform. General requirements for all projects will be discussed in class after the midterm.
- Paper presentation - 10%
Students will present, in groups of two, one research paper during the semester. These papers cover very recent developments in cloud computing. The presentations (using power point slides) will take place in class, and extra-credit will be assigned for active participation in discussions.

**Course Policies**
Use your NJIT email address to send emails and include “CS643” in the subject line. Your emails may be filtered out if you don’t follow.

Visit the course homepage on Moodle regularly and frequently for lecture notes, assignments, instructions, and latest updates.

For assignments and the project, all your work must be submitted electronically via the course Moodle page. Keep your work brief and to the point. You may discuss the questions in general terms with your classmates, but you must come up with your own solutions and all written work must be your own. Copying programs or written assignments (from any source, no matter your classmates or internet) is a violation of NJIT honor code.

As a general rule, no late submission will be accepted, no makeup exams will be given, and no alternate dates for exams without a legitimate reason (e.g., jury duty, medical problem). The legitimacy is determined by the office of the Dean of Students.

Collaboration of any kind is not allowed in any exams. You are not allowed to take the exam of another section. Students with special needs are advised to make arrangements with Disability Services for exam accommodations.

**Honor Code**
Students must follow The NJIT Honor Code. Any violations will be brought to the immediate attention of the Dean of Students.
Fall 2015
CS 645 – Security and Privacy in Computer Systems

Section: 101
Instructor: Reza Curtmola
Office: GITC 4301
Office phone: 973-596-5776
Fax: 973-596-5777
Email: crix@njit.edu
Course website: http://web.njit.edu/~crix/CS.645
Instructor website: http://web.njit.edu/~crix

Office hours: M 4-5 pm, W 4-5 pm. Also, by appointment via email.

Prerequisites:
Students are expected to enter this course with a basic knowledge of operating systems, networking, algorithms, and data structures. Also, students should be able to program in Java and C/C++ for the programming component of the mini-projects.

Textbook:

In addition, course material will include research articles from electronic databases such as: ACM Digital Library (http://dl.acm.org), IEEE Xplore (http://ieeexplore.ieee.org), and Science Direct (http://www.sciencedirect.com)

Description:
The course covers fundamental principles of building secure systems and techniques to protect data privacy. Topics include access control mechanisms, operating systems security, malicious code threats and software security, trusted computing, content protection, and database security. The course will also study existing technical approaches to protecting privacy, including Web anonymizers and anti-censorship tools, as well as policy and legal aspects of privacy.

A tentative list of topics includes:
- Introduction (security goals, overview of course topics, overview of attacks)
- Crypto crash course
- Access control mechanisms
- Operating systems security
- Software security, Secure Programming
- Malicious code, Malware, Rootkits
- Trusted computing
- Introduction to security of networked systems
• Privacy and anonymity on the Web
• Content protection, Software obfuscation, Digital rights management
• Database security
• Security of electronic voting
• Computer crime - laws and ethics, Security & privacy policy (Sarbanes Oxley, HIPAA)
• Miscellaneous topics: side-channel attacks, gaming security, information assurance (common criteria), risk analysis

Grading:
3 Projects: 45%
Midterm exam: 25%
Final exam: 30%

Extra credit will be given for active participation in discussions during the class (up to 10%). The quizzes and exams are closed book unless specified otherwise.

Learning Goals:
• Get familiar with the main types of attacks that may occur in computer systems and networks.
• Understand the various models to evaluate the security of computer systems. Interpret security guarantees. Assess the level of security provided by a cryptographic protocol.
• Understand the various security goals for communication over an insecure network. Identify the appropriate types of cryptographic primitives that should be used to achieve each of these goals. Understand the advantages and limitations of using symmetric-key versus public-key cryptography.
• Get familiar with various types of authentication mechanisms in computer systems and networks.
• Understand the role of physical security in securing computer systems and networks.
• Become familiar with mechanisms used to secure major operating and file systems.
• Understand the main types of attacks against computer programs and learn the principles of secure programming.
• Learn the main mechanisms to secure the communication between computer systems at various network layers.
• Understand the main types of attacks against web-based system and learn the principles of building secure web-based systems.
• Design and implement new security frameworks in order to achieve security guarantees for a diverse set of real-world scenarios.
• Apply theoretical concepts in practice by using a programming or scripting language to implement attacks and defenses against web-based systems.
Honor Code:
The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students. Note in particular that copying lab assignments or exam papers, in full or in part is forbidden.

 Modifications to Syllabus:
The syllabus may be modified at the discretion of the instructor or in the event of extenuating circumstances. Students will be notified in class of any changes to the syllabus.
Computer Networks - Architectures, Protocols and Standards

Course No.
CS 652

Section
101

Title
Computer Networks – Architectures, Protocols and Standards

Course Website
http://cs.njit.edu/~karvelas/CS-652-F15

Prerequisite(s)
CS-356 or some other undergraduate course in the area of networking

Instructor
Dennis Karvelas
Office Room No. : GITC 4212
Office Phone : 973-596-2987
Fax : 973-596-5777
Email : dionissios.karavelas@njit.edu
Website: http://cs.njit.edu/~karvelas

Instructor Office Hours
Mondays: 5.10 pm - 5.50 pm
Tuesdays: 4.30 pm - 5.50 pm
Thursdays: 5.10 pm - 5.50 pm

Description
This course provides an in-depth study of a wide variety of networking technologies as well as hands on experience on network design and traffic analysis. Topics covered include: protocol architectures, multiplexing and its applications, local and wide area landline technologies, last mile broadband access technologies, wireless local area networks, internetwork quality of service, multiprotocol label switching, software defined networks.

Grading Scheme
Two Quizzes:  8% each
Two assignments: 8% each
Project: 10%
Midterm: 26%
Final: 32%
Topics
1. Introduction to Data Communication Networks and the Internet
2. Protocol Architecture, Internet-based Applications
3. Multiplexing Techniques
4. Last Mile Broadband Access
5. Wide Area Network Technologies and Protocols
6. Local Area Networks (LANs)
7. Ethernet Generations
8. Wireless Local Area Networks (WLANs)
9. Internetwork Quality of Services
10. Multiprotocol Label Switching (MPLS)
11. Software Defined Networks (SDN)

Learning Goals and Measurable Learning Outcomes
1. Understand the forces driving the architecture and evolution of networking technologies.
2. Understand the fundamental differences between network architectures and the reasons behind these differences.
3. Gain in-depth understanding of a wide variety of high speed wired and wireless local and wide area network technologies.
4. Learn the broadband last mile technologies being deployed to provide high speed access to companies and residential users.
5. Understand the concepts of the Integrated Services Architecture.
7. Understand the Multiprotocol Label Switching (MPLS) operation, MPLS traffic engineering, and MPLS-enabled VPNs.
8. Gain hands-on experience in network design and analysis
9. Critically analyze a published article in the area.

Text Book(s)
Some of the material will be based on published articles.

Time & Place
Mondays, 6.00 pm – 9.05 pm; KUPF 203

Other Info
Honor Code:
The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students. Note in particular that copying programming assignments or exam papers, in full or in part, is forbidden.
Computer Networks - Architectures, Protocols and Standards

Course No.
CS 652

Section
103

Title
Computer Networks – Architectures, Protocols and Standards

Course Website

Prerequisite(s)
CS-356 or some other undergraduate course in the area of networking

Instructor
Dennis Karvelas
Office Room No.: GITC 4212
Office Phone: 973-596-2987
Fax: 973-596-5777
Email: dionissios.karvelas@njit.edu
Website: http://cs.njit.edu/~karvelas

Instructor Office Hours
- Mondays: 5.10 pm - 5.50 pm
- Tuesdays: 4.30 pm - 5.50 pm
- Thursdays: 5.10 pm - 5.50 pm

Description
This course provides an in-depth study of a wide variety of networking technologies as well as hands on experience on network design and traffic analysis. Topics covered include: protocol architectures, multiplexing and its applications, local and wide area landline technologies, last mile broadband access technologies, wireless local area networks, internetwork quality of service, multiprotocol label switching, software defined networks.

Grading Scheme
Two Quizzes: 8% each
Two assignments: 8% each
Project: 10%
Midterm: 26%
Final: 32%
Topics
1. Introduction to Data Communication Networks and the Internet
2. Protocol Architecture, Internet-based Applications
3. Multiplexing Techniques
4. Last Mile Broadband Access
5. Wide Area Network Technologies and Protocols
6. Local Area Networks (LANs)
7. Ethernet Generations
8. Wireless Local Area Networks (WLANs)
9. Internetwork Quality of Services
10. Multiprotocol Label Switching (MPLS)
11. Software Defined Networks (SDN)

Learning Goals and Measurable Learning Outcomes
1. Understand the forces driving the architecture and evolution of networking technologies.
2. Understand the fundamental differences between network architectures and the reasons behind these differences.
3. Gain in-depth understanding of a wide variety of high speed wired and wireless local and wide area network technologies.
4. Learn the broadband last mile technologies being deployed to provide high speed access to companies and residential users.
5. Understand the concepts of the Integrated Services Architecture.
7. Understand the Multiprotocol Label Switching (MPLS) operation, MPLS traffic engineering, and MPLS-enabled VPNs.
8. Gain hand-on experience in network design and analysis
9. Critically analyze a published article in the area.

Text Book(s)
Some of the material will be based on published articles.

Time & Place
Thursdays, 6.00 pm – 9.05 pm; FMH 408

Other Info
Honor Code:
The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students. Note in particular that copying programming assignments or exam papers, in full or in part is forbidden.
Computer Networks - Architectures, Protocols and Standards

Course No.
CS 652

Section
851

Title
Computer Networks – Architectures, Protocols and Standards

Course Website
njit2.mrooms.net

Prerequisite(s)
CS-356 or some other undergraduate course in the area of networking

Instructor
Dennis Karvelas
Office Room No. : GITC 4212
Office Phone : 973-596-2987
Fax : 973-596-5777
Email : dionissios.karvelas@njit.edu
Website: http://cs.njit.edu/~karvelas

Instructor Office Hours
Mondays:  5.10 pm - 5.50 pm
Tuesdays: 4.30 pm - 5.50 pm
Thursdays: 5.10 pm - 5.50 pm

Description
This course provides an in-depth study of a wide variety of networking technologies as well as hands on experience on network design and traffic analysis. Topics covered include: protocol architectures, multiplexing and its applications, local and wide area landline technologies, last mile broadband access technologies, wireless local area networks, internetwork quality of service, multiprotocol label switching, software defined networks, named data networking architecture.

Grading Scheme
Two Quizzes: 8% each
Two assignments: 8% each
Project: 10%
Midterm: 26%
Final: 32%
Topics
1. Introduction to Data Communication Networks and the Internet
2. Protocol Architecture, Internet-based Applications
3. Multiplexing Techniques
4. Last Mile Broadband Access
5. Wide Area Network Technologies and Protocols
6. Local Area Networks (LANs)
7. Ethernet Generations
8. Wireless Local Area Networks (WLANs)
9. Internetwork Quality of Services
10. Multiprotocol Label Switching (MPLS)
11. Software Defined Networks (SDN)
12. Named Data Networking Architecture (NDN)

Learning Goals and Measurable Learning Outcomes
1. Understand the forces driving the architecture and evolution of networking technologies.
2. Understand the fundamental differences between network architectures and the reasons behind these differences.
3. Gain in-depth understanding of a wide variety of high speed wired and wireless local and wide area network technologies.
4. Learn the broadband last mile technologies being deployed to provide high speed access to companies and residential users.
5. Understand the concepts of the Integrated Services Architecture.
7. Understand the Multiprotocol Label Switching (MPLS) operation, MPLS traffic engineering, and MPLS-enabled VPNs.
8. Understand the motivation for new network architectures as well as the architectural principles driving the design of the Named Data Networking architecture.
9. Gain hands-on experience in network design.
10. Gain hands-on experience in analyzing network traffic.
11. Critically analyze a published article in the area.

Text Book(s)

Time & Place
Distance Learning

Other Info
Honor Code:
The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students. Note in particular that copying programming assignments or exam papers, in full or in part is forbidden.
CS656:001 Internetworking & Higher Layer Protocols

Class time  4-5:25PM TR
Classroom   CKB 313

Professor    Dr. Grace Guiling Wang
Email:        gwang@njit.edu
Office:       GITC 4309
Office hour:  3:00-3:50PM TR or by appointment

TA            Yuan Lu
Email:        yl768@njit.edu
Office:       GITC 4325
Office hour:  TBA

Book

Prerequisite
Decent programming skill is a must. CS356 or equivalent courses are suggested.

Course description
This course studies the architecture and protocols of modern computer networks. Topics to be covered include: addressing, routing, transport protocols, flow and congestion control, wireless and mobile networking, web searching and other application protocols. Upon successful completion of the course, students will have gained a deep understanding of the fundamental concepts and principles of designing and implementing modern computer networks.

Course objectives
- Deep understanding of modern computer network architecture
- Broad knowledge on start-of-art network concepts
- Hands-on experience in designing and coding network-related applications
- Improved presentation skill

Policy of Missed Exams
A make-up exam may be taken only after providing written documentation from the Dean of Students.

Honor Code
The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students. Note in particular that copying programming assignments or exam papers, in full or in part is forbidden.
Grading policy

<table>
<thead>
<tr>
<th></th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming project</td>
<td>20%</td>
</tr>
<tr>
<td>Presentation</td>
<td>15%</td>
</tr>
<tr>
<td>Quiz</td>
<td>6%</td>
</tr>
<tr>
<td>Midterm</td>
<td>25%</td>
</tr>
<tr>
<td>Final</td>
<td>30%</td>
</tr>
<tr>
<td>Attendance</td>
<td>4%</td>
</tr>
</tbody>
</table>

The overall score must be higher than 60 to pass the class.

Presentation
There will be one presentation. Students will study a hot networking technology and present it in class. Through the presentation and feedback from peers and the instructor, presenters are expected to improve their public speaking capability and learn how to present technical topics clearly. Other students are expected to learn a wide range of hot topics.

Programming project
There will be one programming project which helps students gain hands-on experience in designing and coding network-related applications. The project is done in a group. Discussion among classmates is encouraged, but reading and copying each other’s code is forbidden.

Quiz
There will be four quizzes. The one with the lowest score will be discarded and each of the remaining three is 2% of the final grade.

Attendance and in-class performance/behavior
Discussion and questions are highly encouraged. Cell phones must be turned off or set silent during the class hours. Laptop must be turned off. Attendance is required.

Exam
Both the midterm and the final exam are closed-book. The final exam will be comprehensive, but a focus will be the part after the midterm.

Homework
The instructor will assign homework and give solution. But the homework will not be accounted into the final grade.

Course outline
- Basic concepts of computer networks
- Application architectures
  - HTTP, SMTP, DNS, P2P, Web searching
- UDP, TCP, congestion control, flow control
- IP, subnet, IPv6
- Routing protocols, Multicast, Broadcast
- Link layer: MAC, ARP, DHCP, etc
- Wireless and mobile networks
  - Wi-Fi, cellular internet access, mobile IP
- Multimedia networks
### Weekly topics

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Due day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week1</td>
<td>Introduction, layering concepts</td>
<td></td>
</tr>
<tr>
<td>Week2</td>
<td>Delay, Application architectures</td>
<td></td>
</tr>
<tr>
<td>Week3</td>
<td>FTP, SMTP , P2P</td>
<td></td>
</tr>
<tr>
<td>Week4</td>
<td>HTTP, DNS, Web searching</td>
<td></td>
</tr>
<tr>
<td>Week5</td>
<td>Transport layer, UDP</td>
<td></td>
</tr>
<tr>
<td>Week6</td>
<td>Flow control, Congestion Control</td>
<td></td>
</tr>
<tr>
<td>Week7</td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>Week8</td>
<td>Midterm</td>
<td></td>
</tr>
<tr>
<td>Week9</td>
<td>Router, IP, IPv6, Routing</td>
<td></td>
</tr>
<tr>
<td>Week10</td>
<td>ARP/RARP, DHCP, MAC</td>
<td></td>
</tr>
<tr>
<td>Week11</td>
<td>Wi-Fi, Sensor Networks, Ad hoc networks, WiMax</td>
<td></td>
</tr>
<tr>
<td>Week12</td>
<td>Cellular Internet Access, mobile IP</td>
<td></td>
</tr>
<tr>
<td>Week13</td>
<td>Multimedia</td>
<td>Programming Project Due</td>
</tr>
<tr>
<td>Week14</td>
<td>Presentation, Review</td>
<td></td>
</tr>
</tbody>
</table>

### Disclaimer

The weekly topics may be modified at the discretion of the instructor or in the event of extenuating circumstances. Students will be notified in class of any changes to the weekly topics.
Internet and Higher Layer Protocols

Course No.
CS 656

Section
101

Title
Internet and Higher Layer protocols

Course Website
http://cs.njit.edu/~karvelas/CS-656-F15

Prerequisite(s)
CS-356 or some other undergraduate course in the area of networking

Instructor
Dennis Karvelas
Office Room No. : GITC 4212
Office Phone : 973-596-2987
Fax : 973-596-5777
Email : dionissios.karvelas@njit.edu
Website: http://cs.njit.edu/~karvelas

Instructor Office Hours
   Mondays: 5.10 pm - 5.50 pm
   Tuesdays: 4.30 pm - 5.50 pm
   Thursdays: 5.10 pm - 5.50 pm

Description
This course provides an intensive study of interconnected networks and the corresponding protocols that make interconnected architectures function as a single unified communication system. Topics covered include Internet and its structure, Network Architectures, Performance Parameters, Application Protocols (Telnet, FTP, HTTP, SMTP, POP, IMAP, DNS, DHCP), Socket Interface, UDP, TCP, Flow and Congestion Control, IP Addresses, Subnetting, CIDR, Internet Protocol (IP), ICMP, Routing Protocols (RIP, OSPF, BGP, Multicasting), Link Access Technologies (Ethernet, VLANs, ARP, RARP), Wireless and Mobile Technologies (802.11, Bluetooth, Zigbee, 3G, 4G, Mobile IP), Security Issues in Computer Networks.

Grading Scheme
Two Quizzes: 11% each
Project: 12%
Midterm: 33%
Final: 33%
Topics
1. Introduction to Internet and its Architecture, Performance Parameters, Protocols
   Security Issues
2. Telnet, FTP, HTTP, SMTP, P2P File Distribution, DNS, DHCP
3. Socket Programming
4. TCP, UDP, TCP Congestion Control
5. IP and ICMP
6. OSPF, RIP, BGP
7. Link Layer Services, Switched LANs, VLANs, ARP
8. Wireless and Mobile Networks
9. Security in Computer Networks

Learning Goals and Measurable Learning Outcomes
1. Understand the Internet Architecture
2. Gain in-depth understanding of the operation of various application, transport, network
   and link layer protocols
3. Gain hands-on experience in socket programming
4. Gain hands-on experience in traffic analysis by analyzing the traffic of various protocols
   using the Wireshark packet analyzer

Text Book(s)
Some of the material will be based on published articles.

Time & Place
Tuesdays, 6.00 pm – 9.05 pm; KUPF 107

Other Info
Honor Code:
The NJIT Honor Code will be upheld, and any violations will be brought to the
immediate attention of the Dean of Students. Note in particular that copying
programming assignments or exam papers, in full or in part is forbidden.
CS659 - IMAGE PROCESSING AND ANALYSIS  
Fall 2015

INSTRUCTOR: Dr. Frank Y. Shih  
Office: Room 4205, GITC Building  
Office hours: Monday 1:00–2:25 & Thursday 1:45–2:25 in my office (GITC 4205). Also, you can see me by appointment or send me an e-mail. 
Phone: 973-596-5654  
Email: shih@njit.edu

COURSE DESCRIPTION:

This course is an intensive study of the fundamentals of image processing, analysis and understanding. Topics to be covered include: a brief review of the necessary mathematical tools, human visual perception, sampling and quantization, image transformation, enhancement, restoration, compression, reconstruction, image geometric transformation, matching, segmentation, feature extraction, representation and description, recognition and interpretation, and so on.

COURSE OBJECTIVES

The objectives of this course are to: 
1. Cover the fundamental theory and algorithms that are widely used in digital image processing  
2. Expose students to current technologies and applications related to image processing  
3. Develop hands-on experience in using computers to process images  
4. Familiarize with MATLAB Image Processing Toolbox  
5. Develop creative thinking on solving problems of the state-of-the-art in image processing

GRADING:

Homework (20%), Exam 1 (20%), Exam 2 (20%), Paper 1 (20%), Paper 2 (20%)

A. The course schedule (subject to change):

9/13, 11:00PM: Homework 1 Due (Lecture 1, 2, 3)  
9/27, 11:00PM: Homework 2 Due (Lectures 4, 5, 6)  
10/1, 2:00PM–3:30PM, Exam 1; 3:30–5:30 Oral Presentation  
10/11, 11:00PM, Research Paper 1 Due  
10/25, 11:00PM: Homework 3 Due (Lectures 7-9)  
11/8, 11:00PM: Homework 4 Due (Lectures 10-13)  
11/19, 2:00PM–3:30PM, Exam 2; 3:30–5:30 Oral Presentation  
12/6, 11:00PM: Research Paper 2 Due

B. Lecture-3rd Edition Book Correspondance
Notes: Textbook
Lecture 1: Ch 1
Lecture 2: Ch 2
Lecture 3: Ch 4, pp. 199-254
Lecture 4: Ch 3
Lecture 5: Ch 4, pp. 255-310
Lecture 6: Supplement
Lecture 7: Ch 10 & Supplement
Lecture 8: Ch 11 & Supplement
Lecture 9: Ch 9 & Supplement
Lecture 10: Ch 12, pp. 861-882
Lecture 11: Ch 12, pp. 882-902
Lecture 12: Ch 12, pp. 903-906 & Supplement
Lecture 13: Supplement

The following programming languages are allowed: Matlab, C++, and Java. However, Matlab is highly recommended since it offers lots of image-processing functions. It can be easily learned by reading any Matlab programming books. You can download the Matlab software from the NJIT website: http://ist.njit.edu/software/download.php.

There will be a total of 100 points. The grade assign is based on the following:

A: 90 – 100 points
B+: 80 – 89 points
B: 70 – 79 points
C+: 65 – 69 points
C: 60 – 64 points
D: 50 – 59 points
F: 0 – 49 points

COURSE WEBSITE:
Submit the homework solution in Microsoft Word format to http://njit2.mrooms.net/ before the above deadline. Absolutely, no late submission is accepted. Write the answers in your own words individually. Any plagiarism will cause a “FAIL” grade and report to Dean of Students. The course lecture and power-point files are available free of charge through NJIT website at http://njit2.mrooms.net/.

TEXTBOOKS:

REFERENCE BOOKS:

**COURSE CONTENT:**

- Digital Image Fundamentals
- Image Transform
- Image Enhancement
- Image Matching
- Mathematical Morphology
- Image Segmentation
- Image Representation
- Pattern Recognition
- Image Watermarking and Steganography

**Academic Honor Code**

The NJIT academic honor code (http://www.njit.edu/academics/honorcode.php) applies in full to this class. Note in particular that copying programs, in full or in part, is forbidden. You may discuss ideas and concepts with your fellow students, but you may NOT copy any code.
CS 656: Internet and Higher Layer Protocols, Fall 2015
Instructor
George Blank
Office: GITC 4404
NJIT Phone: (973) 596-5485 Home Office (973) 625-0803
Office Hours: Monday 3:00-4:30 pm; Tuesday 1-2:15 and Friday 10:30-11:45am.

Goals
In-depth study of the Internet architecture and protocols
Apply what you learned in a semester-long programming project
Understand how networking research is done by reading and presenting research papers
Short Description
This course studies the architecture and protocols of the Internet. Topics to be covered include:
addressing, routing, transport protocols, flow and congestion control, wireless and mobile networking,
P2P, multimedia, security, web searching and other application layer protocols. To acquire practical
experience, students will implement several features of Internet protocols in a semester-long
programming project. Additionally, students will read and present research papers for a deeper
understanding of networking concepts.

Prerequisites
CS 356 or equivalent. Students are expected to be capable of programming in either Java (used for code
examples in this class) or C/C++ (acceptable, but only minimally supported through examples in this
class) in order to complete the programming project.

Book
by James F. Kurose and Keith W. Ross
Publisher: Addison Wesley

Schedule
Week Topic Readings
2 DNS. The client-server model. Socket programming. First phase of the project handed out.
3 E-mail. HTTP and Web Searching.
4 P2P Networks.
5 UDP. Principles of Reliable data transfer. First phase of the project due. Second phase of the project
handed out.
6 TCP. Congestion control.
7 Midterm. Discussion of midterm solutions.
8 IP, ICMP. Routing I.
9 Routing II. Second phase of the project due. Third phase of the project handed out.
10 Multicast, Broadcast, QoS in Internet.
11 Data Link Layer: ARP/RARP, DHCP, and MAC.
12 Wireless and mobile networks: Wi-Fi, Cellular Internet Access, Mobile IP.
13 Multimedia: RTP, RTSP, MPEG.
14 Network Security: authentication, attacks, access control. Third phase of the project due.

Lecture Notes and Assigned Readings
The lecture slides will be posted on the class webpage in Moodle prior to the lecture. Students are required to read the assigned book chapters before each lecture.

Exams
There will be three exams: two midterms, and a final exam. Exams are closed book/notes. The final exam will cover only the material taught after the midterm.

Programming Project
Students will work individually toward implementing a micro-Internet at the application layer. This micro-Internet will emulate transport layer features (e.g., reliability, congestion control) and network layer features (e.g., forwarding and routing). The project will be divided in three phases, which will be graded separately. Solutions for the intermediate phases will be posted immediately after each submission; in this way, students can work on a new phase even if they didn’t implement completely the previous phase. The submissions will be done electronically. Test cases will be posted one week before the due dates. Late submissions will not be accepted.

Homework
Homework will be assigned weekly to prepare students with the type of questions encountered in exams. The homework consists of a few questions selected/edited by the instructor and posted by midnight on Fridays. The homework is due at the next class. Late submissions will not be accepted. Students will be assigned to present solutions in class. In addition, there are 8 Wireshark Labs that must be turned in.

Reading and Presenting Research Papers
Ph.D students and student’s seeking grades above B are required to read and present one research paper during the semester. In this way, students will learn fundamental lessons from the design of the Internet and be exposed to recent developments in the networking area. As byproducts, students will understand how networking research is done and learn to speak in public about a research topic. The papers will be selected by the instructor from top conferences and journals. The presentations (using power point slides) will take place in class before every lecture (starting with the 3rd week). Extra-credit will be assigned for active participation in the discussions.

Grading
Each Midterm - 25%
Final exam - 25%
Programming project - 15%
Research paper and homework presentation - 10%
Policy of Missed Exams
A make-up exam may be taken only after providing written documentation from the Dean of Students.

Honor Code
The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students. Note in particular that copying homework or programming assignments, in full or in part, is forbidden.

Modifications to Syllabus
The students will be consulted and must agree to any modifications or deviations from the syllabus throughout the course of the semester.
New Jersey Institute of Technology
Department of Computer Science

CS670 - Artificial Intelligence - Fall'2015

Tuesday 6:00 - 9:05 PM, CKB 214

Course Description | Outcomes | Readings | Tentative Contents | Grading Policy | Miscellaneous

Chengjun Liu, Ph.D.

Phone: 973-596-5280
Email: chengjun.liu@njit.edu
Office: GITC 4306

Office Hours: Tuesday & Wednesday 3:30-5:00PM or by appointment
(caveat: office hours on W's, Sep. 16 - Oct. 7, are moved to T's 2:00-5:00PM)

Course Description

- This course introduces concepts, approaches and techniques of artificial intelligence, and focuses on materials that are fundamental and have a broad scope of applications. Topics include Problem Solving, Intelligent Agents, Knowledge and Reasoning, Logical Agents, Propositional Logic, First-Order Logic, Uncertain Knowledge and Reasoning, Quantifying Uncertainty, Probabilistic Reasoning, Learning, Statistical Learning Theory, Bayesian Learning, Decision Tree learning, Neural Networks, Genetic Algorithms, Reinforcement Learning, Perception, Pictorial Knowledge Representation, Search, Action, and Robotics.

- Prerequisite: CS 610 - Data Structures and Algorithms

Measurable Learning Outcomes

- Students learn the concepts, approaches and techniques of artificial intelligence.
- Students learn the materials that are fundamental and have a broad scope of applications in artificial intelligence, such as Problem Solving, Intelligent Agents, Knowledge and Reasoning, Logical Agents, Propositional Logic, First-Order Logic, Uncertain Knowledge and Reasoning, Quantifying Uncertainty, Probabilistic Reasoning, Learning, General Learning Model, Decision Tree learning, Unsupervised Learning, Supervised Learning, Statistical Learning Theory, Structural Risk Minimization, Support Vector Machine, Perception.

Readings

• Selected papers and handouts.

**Tentative Contents**

1. Introduction
   - AI Fundamentals (Turing test, cognitive science, logic, learning, games, robot, vehicle, agent)
   - AI prehistory and AI history: connectionism, symbolism, AI winters
   - Programming Languages: Lisp, Prolog, Matlab, C/C++, Java
   - Related Fields: ML, NN, EC, CV, PR, IP
2. Problem Solving
   - Intelligent Agents
   - Solving Problems by Searching
   - Breadth-first Search, Depth-first Search
   - Best-first Search, Greedy Search, A* Search
   - Games (Adversarial Search, Alpha-Beta Pruning)
3. Knowledge and Reasoning - Logical Agents
   - Knowledge-Based Agents
   - Logic, Propositional Logic
   - Models, Semantics, Inference, Validity and Satisfiability
   - Propositional Theorem Proving, Resolution, CNF
   - Games (Wumpus World)
4. Knowledge and Reasoning - First-Order Logic
   - FOL Syntax and Semantics
   - FOL Sentences, Models, Interpretation
   - FOL Quantification, Properties of Quantifiers
   - FOL KBs, Deducing Hidden Properties
5. Knowledge and Reasoning - Inference in First-Order Logic
   - Propositional vs. First-Order Inference
   - Universal and Existential Instantiation
   - Unification, GMP, Soundness of GMP
   - FOL KB and Resolution
   - Logic Programming - Prolog
6. Uncertain Knowledge and Reasoning - Quantifying Uncertainty
   - Acting under Uncertainty
   - Uncertainty and Probability
   - Syntax and Semantics
   - Inference by Enumeration, Normalization
   - Independence, Conditional Independence, Bayes' Rule
7. Uncertain Knowledge and Reasoning - Probabilistic Reasoning (optional)
   - Bayesian Networks
   - Hidden Markov Models
   - Kalman Filters
8. Learning - Theory of Learning
   - General Learning Model
   - Inductive Learning
   - Learning Decision Trees
9. Learning - Unsupervised Learning
   - Principal Component Analysis
   - Applications: Compression, Feature Representation
10. Learning - Supervised Learning
    - Discriminant Analysis
    - Applications: Feature Extraction for Classification
11. Learning - Probabilistic Models
    - Statistical Learning Theory (STL)
    - Structural Risk Minimization (SRM)
    - Support Vector Machines (SVM)
12. Learning - Other Popular Models (optional)
    - The EM Algorithm
    - Bayesian Learning
    - Genetic Algorithms
    - Reinforcement Learning
13. Perception - Search in Spatial Domain and Frequency Domain (optional)
    - FFT, Lowpass and Highpass Filtering, Convolution Theorem
    - Edge Detection, Line and Curve Detection (Hough Transform)
    - Pictorial Information Search using Geometric or Frequency Features
    - Sensors and Vision
    - Path Planning
    - Moving and Control

Grading Policy

- Homework 20%
- Midterm exam 20%
- Project and presentation (topics are related to our course Contents) 20%
- Class attendance and participation 10%
- Final exam 30%

Academic integrity and honesty are of paramount importance. NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students.

Miscellaneous

- Berkeley AI Course Materials
- Prolog:
  - J.R. Fisher, *The Prolog Tutorial*
- Lisp:
Syllabus: CS 673, Fall 2015

Software Design and Production Methodology

1. Course Description

CIS 673 - Software Design and Production Methodology (3 credits)  
Prerequisite: CIS 631. Modern techniques and methods employed in the development of large software systems, including a study of each of the major activities occurring during the lifetime of a software system, from conception to obsolescence and replacement. Topics include cost/performance evaluation, documentation requirements, system design and production techniques, system verification techniques, automated aids to system development, and project organization and management.

2. Personnel

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ali Mili, Office: GITC 4305</td>
<td></td>
</tr>
<tr>
<td>Fax: (973) 596 5777</td>
<td>Homepage: <a href="http://www.cs.njit.edu/~mili">http://www.cs.njit.edu/~mili</a></td>
</tr>
<tr>
<td>Phone: (973) 596 5215</td>
<td>Email: <a href="mailto:mili@cis.njit.edu">mili@cis.njit.edu</a></td>
</tr>
</tbody>
</table>

3. Readings


*Reuse Based Software Engineering: Techniques, Organization and Measurement.*  

Klaus Pohl, Guenther Boeckle and Frank J. Linden. Springer Verlag, 2005.

*Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering.*  

The students are also expected to read research materials or other documentation as it becomes necessary in the course of the semester.
4. Assignments:

This course revolves around a *Product Line Engineering* project that students carry out throughout the semester. As such, the project includes two phases:

- *Domain Engineering*,
- *Application Engineering*.

Domain engineering will take approximately 13 weeks, and will be carried out in teams. Application engineering will take the remaining two weeks of the semester, and will be done individually.

5. Grading:

Students will be evaluated on the basis of their team work and on the basis of individual work. Team Work includes a lecture given by the team as well as domain engineering deliverables. Individual work includes a mid-term examination, a final examination, and the individual application engineering deliverable.

<table>
<thead>
<tr>
<th>Team Effort</th>
<th>Team Lecture</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Engineering</td>
<td>400</td>
<td></td>
</tr>
<tr>
<td>Individual Effort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Engineering</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Mid-Term</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Final Test</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

Students will be required to submit a set of team deliverables as part of the domain engineering phase. For each phase, teams are expected to produce a word file that contains the text of their deliverable. The team that produces the best deliverable may be asked to make a presentation of their deliverable, as a sample deliverable.

6. Web Support:

This course will be delivered with the support of moodle. Visit [http://njit2.mrooms.net/](http://njit2.mrooms.net/) and make sure that the system recognizes you and that it lists this course for you. We will use moodle to post all relevant material; students may use moodle to discuss topics of general interest to the class.

7. Outline of the Course

The schedule of the course will run in parallel with the schedule of the project, and will help prepare students to produce the project deliverables.
<table>
<thead>
<tr>
<th>Wk#</th>
<th>Week</th>
<th>Lecture</th>
<th>Project Deliverable</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sept 8th</td>
<td>Introduction, discussion</td>
<td>Team Formation</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sept 14</td>
<td>Product Line Eng.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Sept 21</td>
<td>Domain Analysis, Methodologies</td>
<td>FAST report</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Sept 28</td>
<td>Software Specifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Oct 5th</td>
<td>Software Qualities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Oct 12</td>
<td>Software Architecture</td>
<td>Domain Analysis</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Oct 19</td>
<td>Soft Eng Economics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Oct 26</td>
<td>Mid Term Examination</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Nov 2nd</td>
<td>Software Reuse Costs</td>
<td>Ref. Architecture</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Nov 9</td>
<td>Product Line Testing</td>
<td>Cost Estimation</td>
<td>TBD</td>
</tr>
<tr>
<td>11</td>
<td>Nov 16</td>
<td>TBD/ TBD</td>
<td></td>
<td>TBD</td>
</tr>
<tr>
<td>12</td>
<td>Nov 23</td>
<td>TBD/ TBD</td>
<td></td>
<td>TBD</td>
</tr>
<tr>
<td>13</td>
<td>Nov 30</td>
<td>TBD/ TBD</td>
<td>Domain Engineering</td>
<td>TBD</td>
</tr>
<tr>
<td>14</td>
<td>Dec 7th</td>
<td>TBD/ TBD</td>
<td></td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td>Dec 13</td>
<td>Application Engineering due, 11:55 PM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dec 14</td>
<td>AE Demo, 9:00 AM – 1:00 PM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dec 21</td>
<td>Final Exam, 6:00 PM – 8:00 PM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Starting in November, student teams will be asked to give lectures on topics related to the theme of the course taken from the most recent SPLC proceedings (splc.net) as well as related sites (e.g. SPL Hall of Fame). In light of the class enrollment, I anticipate to have nine teams of four students each. Student talks will be graded on the basis of the following criteria:

- **Breadth of research:** 30%.
- **Quality of the synthesis:** 45%.
- **Quality of the presentation:** 15%.
- **Quality of the delivery:** 10%.

Teams must be formed and their topic selected by the second lecture of the semester, i.e. by September 14. All students must attend all lectures and make every effort to understand them; also, students are encouraged to ask questions or challenge the speakers if they so feel; at the end of the semester, the final exam will include one question on each student lecture.

Topics of interest may include, but are not limited to:

- Software Product Line Architectures
- Domain Analysis Methodologies
- Feature Modeling
- Domain Modeling
- Product Line Economics
- Product Line Hall of fame
- Service Oriented Product Lines
- Variability Modeling and Representation
- Software Product Line Practice
• Product Line Testing
• Product Line Requirements Engineering
• Application Engineering
• Formal Methods in Software Product Lines

Select your topic in coordination with the instructor, and discuss with him what bibliographic references you will use for your talk.

8. Academic Integrity

Please be advised that academic integrity and honesty are of paramount importance to me. I will uphold NJIT’s honor code, and will have the duty to report any violation of this code to the immediate attention of the Dean of Students.

9. Attendance

Attendance will be recorded at every lecture. Attendance and class participation are greatly encouraged, and adequately rewarded. Please do not sign your name as present unless you attend the whole lecture.

Have a fruitful/ enjoyable semester!
**Professor**  Iulian Neamtiu  
email: iulian.neamtiu@njit.edu  
Office hours: Mondays & Thursdays 2:30p.m. – 3:30p.m.  
GITC 4108

**Textbook**  
Fundamentals of Software Engineering (2nd edition)  
Authors: Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli  

**Prerequisites**  
CS 631  
Credits  
3  
Class  
Mondays & Thursdays, 1:00p.m. – 2:25p.m.  
KUPF 208

**Holidays**  
No class Thu., November 26 (Thanksgiving)

**Catalog description**  
Modern techniques and methods employed in the development of large software systems, including a study of each of the major activities occurring during the lifetime of a software system, from conception to obsolescence and replacement. Topics include cost/performance evaluation, documentation requirements, system design and production techniques, system verification techniques, automated aids to system development, and project organization and management.

**Class website**  
All class interactions (grades, slides, practice material, documentation turn-in, announcements) will take place on Moodle.

### Lecture schedule (subject to minor changes)

<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>Topic</th>
<th>Reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thu</td>
<td>Sept 3</td>
<td>Introduction; Project guidelines</td>
<td>Chapter 1</td>
</tr>
<tr>
<td>Tue</td>
<td>Sept 8</td>
<td>Software: Its Nature and Qualities</td>
<td>Chapter 2</td>
</tr>
<tr>
<td>Thu</td>
<td>Sept 10</td>
<td>Software Engineering Principles</td>
<td>Chapter 3</td>
</tr>
<tr>
<td>Mon</td>
<td>Sept 14</td>
<td>The Software Production Process</td>
<td>Chapter 7 (sections 7.1–7.6)</td>
</tr>
<tr>
<td>Thu</td>
<td>Sept 17</td>
<td>Rapid Development and Extreme Programming</td>
<td>Rapid Development (PDF on Moodle)</td>
</tr>
<tr>
<td>Mon</td>
<td>Sept 21</td>
<td>Industry Practices</td>
<td>How Microsoft Builds Software (available for free from the campus network)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The Facebook Release Process (video)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Exclusive: a behind-the-scenes look at Facebook release engineering (article)</td>
</tr>
<tr>
<td>Thu</td>
<td>Sept 24</td>
<td>Management of Software Engineering</td>
<td>Chapter 8 (sections 8.1–8.2 up to, and including 8.2.3.1), and 8.3)</td>
</tr>
<tr>
<td>Mon</td>
<td>Sept 28</td>
<td>Specification</td>
<td>Chapter 5 (sections 5.1–5.5.1)</td>
</tr>
<tr>
<td>Thu</td>
<td>Oct 1</td>
<td>Finite State Machines; Petri Nets</td>
<td>Chapter 5 (sections 5.5.3 and 5.5.4)</td>
</tr>
<tr>
<td>Mon</td>
<td>Oct 5</td>
<td>Petri Nets (cont’d)</td>
<td></td>
</tr>
<tr>
<td>Thu</td>
<td>Oct 8</td>
<td>First-order Logic</td>
<td>First-order Logic primer (PDF on Moodle)</td>
</tr>
<tr>
<td>Mon</td>
<td>Oct 12</td>
<td>Logic Specification</td>
<td>Section 5.6.2 (up to, and including, 5.6.2.2)</td>
</tr>
<tr>
<td>Thu</td>
<td>Oct 13</td>
<td>Midterm review</td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>Oct 19</td>
<td>Midterm</td>
<td></td>
</tr>
<tr>
<td>Thu</td>
<td>Oct 22</td>
<td>Design and Software Architecture</td>
<td>Chapter 4 (sections 4.1–4.2.8)</td>
</tr>
<tr>
<td>Mon</td>
<td>Oct 26</td>
<td>Design Notation, Abstraction and Refinement</td>
<td>On the Criteria To Be Used in Decomposing Systems into Modules by D. Parnas (available for free from the campus network)</td>
</tr>
<tr>
<td>Thu</td>
<td>Oct 29</td>
<td>Verification; Introduction</td>
<td>Chapter 6 (up to 6.3.4.1)</td>
</tr>
<tr>
<td>Mon</td>
<td>Nov 2</td>
<td>Structural (White-box) Testing</td>
<td>Section 6.3.4 (up to, and including, 6.3.4.1)</td>
</tr>
<tr>
<td>Thu</td>
<td>Nov 5</td>
<td>Functional (Black-box) Testing in the large</td>
<td>Section 6 (sections 6.3.4.2–6.3.5.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Notes on equivalence class testing (PDF on Moodle)</td>
</tr>
<tr>
<td>Mon</td>
<td>Nov 9</td>
<td>Verification: Program Analysis</td>
<td>Section 6.4 (omit 6.4.2.2 and 6.4.2.3); An Axiomatic Basis for Computer Programming by C.A.R. Hoare (available for free from the campus network)</td>
</tr>
<tr>
<td>Thu</td>
<td>Nov 12</td>
<td>Program Analysis (cont’d)</td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>Nov 15</td>
<td>Symbolic Execution</td>
<td>Sections 6.5 (omit 6.5.2) and 6.7</td>
</tr>
<tr>
<td>Thu</td>
<td>Nov 19</td>
<td>Tools and Environments</td>
<td>Section 9.3</td>
</tr>
<tr>
<td>Mon</td>
<td>Nov 23</td>
<td>Buffer</td>
<td></td>
</tr>
<tr>
<td>Thu</td>
<td>Nov 26</td>
<td>No class (Thanksgiving)</td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>Nov 30</td>
<td>Review</td>
<td></td>
</tr>
<tr>
<td>Thu</td>
<td>Dec 3</td>
<td>Project presentations</td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>Dec 7</td>
<td>Project presentations</td>
<td></td>
</tr>
<tr>
<td>Thu</td>
<td>Dec 10</td>
<td>Project presentations</td>
<td></td>
</tr>
</tbody>
</table>
Students are expected to attend, follow, and actively participate in all lectures and discussions. All lectures and exams start at the stated time. Avoid being late coming to class, as this is very disruptive. Students are requested to refrain from using cell phones and iPod-like devices during lectures. If, for some compelling reason (e.g., a member of your family is in the hospital), you need to receive calls, be sure to put your cell phone on silent, and excuse yourself from the class if you need to take a call. Laptops and tablets are permitted for taking notes. Recording the lecture (audio or video) is prohibited. No electronic devices are permitted during exams.

Academic integrity
Academic integrity will be strictly enforced. Any violation or suspected violation of academic integrity will be dealt with according to NJIT’s University Code on Academic Integrity. 

Ignorance is no excuse.

<table>
<thead>
<tr>
<th></th>
<th>Percent Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>15</td>
</tr>
<tr>
<td>Project</td>
<td>65</td>
</tr>
<tr>
<td>Final exam</td>
<td>20</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

Assuming your overall numerical grade is \( x \), your letter grade is:

\[
\begin{align*}
    x < 63 & : F \\
    63 \leq x < 73 & : C \\
    73 \leq x < 77 & : C+ \\
    77 \leq x < 83 & : B \\
    83 \leq x < 87 & : B+ \\
    87 \leq x & : A
\end{align*}
\]

This scheme operates on real numbers hence there will be no rounding, up or down.

Project Guidelines

In this class you will complete a substantial team project. The project score is 65% of your grade. Teams will consist of 3–4 students; team composition is part voluntary, part randomly assigned.

Team Assignment
You have until September 7@5:00p.m. (see Table 1) to form up a voluntary team core, of at most 3 members as follows: one of the team core members must send the team core composition by email to the professor and cc the rest of the team core members. Team core composition is not official until confirmed by email by the professor—it is your responsibility to ensure you have a confirmed team core by the team core deadline. The rest of the team (if any, up to 4 members) is assigned by the professor. If, by September 7, you do not belong to any team core yet, you will be randomly assigned to a team.

Project Assignment
After your team composition is known (on September 9@5:00p.m. at the latest), the professor will disclose the projects list. You will have two days (until September 11@5:00p.m.) to send the professor a ranking of all projects in order of preference. The professor will then assign a project to each team; we will do our best to assign you a project toward the top of your list, but it is not guaranteed that you will get your #1 choice. If, by September 11@5:00p.m., you will not have sent us a ranking list, you will be randomly assigned one of the unclaimed projects.

Projects and Milestones
Each project’s requirements will be split into two sets: Milestone 0 and Milestone 1. Milestone 0 requirements are supposed to be completed by the mid-quarter assessment. Milestone 1 requirements are supposed to be completed by the final assessment. Any Milestone 0 requirements not fulfilled by the mid-quarter assessment will impact your mid-quarter score negatively and will still be required for the final assessment.

Project Documentation
For both the mid-quarter and final assessments, you must attach a document describing the project. The professor will provide the format of this document; mandatory sections will include requirements, design, operating manual, etc.

Teamwork ethics
As in all team-oriented coursework, be especially aware that you have a responsibility (not only toward yourself and the instructor, but also toward the other members of your team) to perform the required work in a professional and timely manner. Past experience suggests that a good team dynamic (which includes frequent team meetings, constant awareness of “who-does-what,” and a collaboration infrastructure to enable that, i.e., version control, feature/bug tracking) is just as essential as individual efforts to the successful completion of the project. To that end, each team member must act responsibly by checking email frequently, replying promptly, and making every effort to attend team meetings.

1. One of the team members must send the list by email to the professor; cc the rest of the team members as well. A team’s project ranking list is not official until confirmed by email by the professor.
Table 1: Important dates.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Day</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept 7</td>
<td>5:00 p.m.</td>
<td>Mon</td>
<td>Voluntary team assignment due</td>
</tr>
<tr>
<td>Sept 9</td>
<td>5:00 p.m.</td>
<td>Wed</td>
<td>Teams assigned</td>
</tr>
<tr>
<td>Sept 11</td>
<td>5:00 p.m.</td>
<td>Fri</td>
<td>Project rankings due</td>
</tr>
<tr>
<td>Sept 14</td>
<td>5:00 p.m.</td>
<td>Mon</td>
<td>Projects assigned</td>
</tr>
<tr>
<td>Oct 19</td>
<td>in class</td>
<td>Thu</td>
<td>Midterm</td>
</tr>
<tr>
<td>Oct 29</td>
<td>5:00 p.m.</td>
<td>Thu</td>
<td>Mid-quarter (Milestone 0) assessment due: milestone requirements, documentation, individual contribution sheet</td>
</tr>
<tr>
<td>Nov 2</td>
<td>5:00 p.m.</td>
<td>Mon</td>
<td>Peer feedback due</td>
</tr>
<tr>
<td>Dec 3</td>
<td>5:00 p.m.</td>
<td>Thu</td>
<td>Final assessment (Milestone 1) due: milestone requirements, documentation, individual contribution sheet</td>
</tr>
<tr>
<td>Dec 3</td>
<td>in class</td>
<td>Thu</td>
<td>Project presentations</td>
</tr>
<tr>
<td>Dec 7</td>
<td>in class</td>
<td>Mon</td>
<td>Project presentations</td>
</tr>
<tr>
<td>Dec 10</td>
<td>in class</td>
<td>Thu</td>
<td>Project presentations</td>
</tr>
<tr>
<td>Dec TBD</td>
<td></td>
<td></td>
<td>Final exam</td>
</tr>
</tbody>
</table>

Team work has many advantages (mutual support, best use of individual skills, encouragement, camaraderie), but this is true only if everyone participates actively to the project. To ensure that every team member contributes fairly, each team will provide a status report by email at the beginning of the lab, starting with the fourth week of classes (September 24). The reports, one or two paragraphs per member, must clearly state each team member's contribution to the team's activities.

At the end of each milestone, each team must submit a signed contribution sheet indicating the contribution of each member in percents. The instructor will use these evaluation forms, together with his assessment of students' effort, to determine the grade. All documents must be submitted in PDF format. Except for individual issues, when communicating with the instructors be email, please cc all your teammates.

Weekly Contribution Log

By 5:00 p.m. each Thursday, starting September 24, each team (i.e., one of the team members, cc'ing the teammates) must submit a weekly contribution log by email, to the instructor. The log must contain, for each team member, the progress that member made toward completing the project; details matter, so we expect at least one paragraph per person. Failure to submit a contribution log each week before the beginning of the lab, or submitting an incomplete/uninformative log will result in points being deducted from that team’s project score. The email’s subject should look like this: “CS 673 Weekly Contribution Log: Team 6, week 09/18–09/24”; of course, substitute in the appropriate team number and week.

Individual Contribution Sheet

When an assessment is due (i.e., on October 29 and December 3, respectively), each team must submit an individual contribution sheet in two copies: a signed hard copy and by email (the email copy doesn’t need to be signed; the email’s subject should look like this: “CS 673 Contribution Sheet: Team 6, Milestone 0”; of course, substitute in the appropriate team number and milestone number).

The sheet must explain how the team divided the work, researched, and prepared the assessment. The sheet should clearly reflect the contributions of each team member to the assignment, in percents. An ideal contribution proportion would be $\frac{\text{team size}}{100\%}$ from each member, but we prefer that your analysis be honest, rather than ideal. Each student’s contribution (called Individual Contribution) will be factored into that student’s score.

Each member of the team will indicate at the bottom of the sheet that he or she has read the project documentation and the contribution sheet and agrees with the contents of both.

Each member should be responsible for a part of the project; all members of a team will receive the same number of points (called Team Score) for the contents and quality of the overall project. Only participation points will vary across members of the team. In order to receive all of the participation points, the overall contributions of each member must be equal; see section “How Are Project Scores Computed?” Since each member is required to sign-off on the contribution sheet, teams must prepare and review the sheet before the due date. Again, if a team cannot agree on, or fails to submit a proper contribution sheet for an assessment before the due date, that team’s score will be 0 (zero) for the respective assessment.

Peer Feedback

Part of the mid-quarter assessment score is based on the quality of a peer feedback document. Each team will read, evaluate, and suggest improvements to its peer team’s project documentation. Feedback assignments (e.g., team X evaluates team Y’s documentation) will be disclosed by the professor, shortly after the mid-quarter project documentations are due, i.e., on October 29. The professor will provide tips on how to write an effective peer feedback document.

Milestone Delivery

Each delivery (Milestone 0 and Milestone 1) delivery will consist of two parts:

- **Requirements completion.** Arrange a meeting with the instructor by sending your team’s joint availability at least one week in advance of the milestone deadline. During the meeting you will demo the features required for that milestone. Failure to arrange a meeting will lead to a score of 0 (zero) for that milestone’s requirements part.

- **Documentation.** A single PDF must be turned in on Moodle by the deadline.

In-class Presentation

Part of the final-assessment score is based on the quality of the in-class presentation (which will take place on Dec 3, Dec 7, Dec 9).
Grading
The 65% project score is split as follows:

- 30% for the mid-quarter assessment, due October 29 (peer feedback due November 2, see Table 1).
- 35% for the final assessment, due December 3.

The grading criteria differ for midterm and final assessments.

Mid-quarter assessment grading criteria:
1. Completion of Milestone 0 requirements (80%).
2. Project documentation (15%).
3. Quality of peer feedback (5%).
4. Individual contribution sheet, hard-copy, signed by each member.

Final assessment grading criteria:
1. Completion of Milestone 1 requirements, project documentation (85%).
2. Quality of in-class presentation (15%).
3. Individual contribution sheet, hard-copy, signed by each member.

The absence of a signed individual contribution sheet for a team will automatically get that team a score of 0 (zero) for the respective assessment. If, due to extenuating circumstances, one of the team members cannot physically sign the sheet, the circumstances must be disclosed in advance, and agreed upon by the professor. Note that submitting the contribution sheet (albeit without all signatures) is still mandatory.

How Are Project Scores Computed?
Each student’s mid-quarter and final scores will be computed based on (1) that team’s project score (TeamScore), which is the same for all team members, (2) the student’s contribution to the project (IndividualContribution), and (3) the expected contribution (ExpectedContribution), i.e., \( \frac{100 \times \text{team size}}{100} \). The exact formula is given in Figure 1.

Implementation
Teams are free to complete the project in any programming language(s) the members are most comfortable with/proficient in. Use of existing free or open source software components, libraries, frameworks, toolkits, etc. is permitted as long as this use is prominently disclosed in the project documentation.

Frequently Asked Questions
- **Q (class): Why is some of the lecture material disconnected from the project material?**
  * A: The lecture covers the theory (principles), while the project covers the practice.

- **Q (class): Why does the project require such a significant effort and moving at a fast pace?**
  * A: This class project is a breeze compared to what professional software engineers face.

- **Q (project): What if team members cannot agree on individual contributions?**
  * A: Failing to submit a completed contribution sheet will result in all team members receiving a 0 (zero) for that assessment.

- **Q (project): What if one of the team members is not contributing enough hence dragging the whole team down?**
  * A: Inform the professor ASAP if such a situation occurs. While the score computation formula is designed to account for such cases, early resolution is highly desirable.

- **Q: We sent the instructor an email, but haven’t heard back?**
  * A: Barring unforeseen circumstances, the professor will respond pretty quickly. Unfortunately, we receive a large number of emails that fail to follow procedure. These emails will not be answered. Frequent examples: sending a team core composition or a project ranking without cc’ing all team members; sending a project ranking that does not include all projects. So if you’re not receiving a quick response, please double-check the syllabus and make sure you are following the procedure.

\[
\text{IndividualScore} = \begin{cases} 
0 & \text{if IndividualContribution} = 0 \\
\text{TeamScore} \times \frac{\text{IndividualContribution}}{\text{ExpectedContribution}} & \text{otherwise; at most 100 points}
\end{cases}
\]

Figure 1: Individual score computation for the project part.
COURSE SYLLABUS

CS 696
Network Management and Security
Fall 2015

Instructor: Kurt Rohloff
Grader: TBD

<table>
<thead>
<tr>
<th>Credits:</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section:</td>
<td>101</td>
</tr>
<tr>
<td>Class hours:</td>
<td>Friday 6:00pm-9:05pm</td>
</tr>
<tr>
<td>Lecture Location:</td>
<td>KUPF108</td>
</tr>
<tr>
<td>Course Website:</td>
<td>moodle</td>
</tr>
</tbody>
</table>

Instructor Info:

<table>
<thead>
<tr>
<th>Instructor Office:</th>
<th>4105 GTIC (Computer Science Building)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor Office Hours:</td>
<td>In person: Tuesday 3pm-4pm and by appointment</td>
</tr>
<tr>
<td>Instructor E-mail:</td>
<td><a href="mailto:rohloff@njit.edu">rohloff@njit.edu</a></td>
</tr>
</tbody>
</table>

Grader Info:

We will have a grader, but this person will be working behind the scenes.

1 DESCRIPTION

Thorough introduction to current network management technology and techniques, and emerging network management standards. In-depth study of the existing network security technology and the various practical techniques that have been implemented for protecting data from disclosure, for guaranteeing authenticity of messages, and for protecting systems from network-based attacks. Various types of security attacks (such as intruders, viruses, and worms) will be covered. Conventional Encryption and Public Key Cryptology. Various security services and standards (such as Kerberos, Digital Signature Standard, Pretty Good Privacy, SNMPv2 security facility). Firewalls, IDS’s, VPNs and other protection techniques will be a focus, including privacy protection and circumvention technologies.

2 ORGANIZATION

This is a lecture course with an extensive self-learning experiential component during outside projects and homework. Topics are introduced and presented by the instructor, with a focus on big-picture concepts and practical implications. Student participation during in-class discussion is encouraged.
Students are assigned homework focused on topics discussed in lecture. There will be assigned readings from supplementary materials. The initial lectures are structured and scheduled, but we leave flexibility in the schedule to discuss topics of current interest selected with input from students. There will be a mid-term, a final exam and a course project.

3 COURSE OBJECTIVES

At the end of the course, students are expected to have an understanding of:

1. Major networking protocols (IP, TCP, UDP, DHCP, DNS, etc...), their vulnerabilities and security features.
2. Common vulnerabilities and attacks, including malware.
3. Network analysis and packet sniffing tools.
4. Firewall and circumvention technologies and techniques.
5. Wireless security, including for the major wireless protocols (802.11, Bluetooth, etc...).
6. How cryptography is used at all layers of the network stack.
7. Most common side-channel attacks networks.
8. Identity management techniques and limitations.
9. Attack detection, damage minimization and forensics.
10. Privacy protection.

4 TEXT AND REQUIRED SUPPLIES

Required text: None
Optional Supplemental Texts:

1. Computer Networking: A Top-Down Approach by Kurose and Ross (any edition, but the later the better.)

Additional easily downloadable material will be assigned over the term. References, URLs, or pdf files will be shared as appropriate.

This class and homework will be experiential and experimentally focused. You are expected to have access to machines with Unix/Linux command line with root access, a network and the ability to install software. MacOS may be fine, but your mileage may vary. Virtual machines running on a Windows machine are fine. An Amazon cloud server is fine. AFS access should be okay, but mileage may vary.

Be sure to follow all applicable laws and standard rules of good conduct in your locality, when applicable.

5 EMERGENCY PROCEDURES

Evacuation procedures – see instructions posted in the classroom.
Emergency Aid – dial 911.
6 YOUR IDEAS, EVALUATIONS, ETC.

In general, your ideas, comments, suggestions, questions, grade challenges, etc. are welcome. Your discretion in these matters is expected, however. No part of your grade will be based on anything other than your assignments.

You are encouraged to take advantage of instructor office hours for help with coursework or any other subject connected with the course and your progress.
7 GRADING PLAN

Final grades are weighted as follows:

1. Weekly Homework: 30%
2. Project: 30%
3. Take-home mid-term Exam: 20%
4. Take-home Final Exam: 20%

If you can get your final project activity published in an ACM or IEEE workshop, symposium or conference, your course grade will be an A, unless there are any honor code violations.

All exams, homework and project work will be open-book and open-Internet.

Any violations of the honor code, including plagiarism, copying, mis-representing others' work, or mis-representing your own work will be treated severely, including referral to the Dean of Students.

Please submit all homework and project work through the course moodle site. I generally assign homework so that they are nominally due at the beginning of class. I will generally be pretty lenient about turning in homework late, as long as I receive homework by the course staff begins to grade them. My goal is to return your homework grades to you by the next class meeting.

Homework not turned in by the time the grader begins work will be considered late and 50% of the homework assignment will be deducted if it is submitted later unless it is more than a week late, when it will be marked as 0%.

Exams will be in class, during normally scheduled class hours. Exams will last an hour and I expect to have an abbreviated lecture after the exam.

We will have a course project that is intended to be open-ended. The project is your chance to explore a topic of your interest in depth. I am going to give a lot of leeway to students to identify a project of interest. Acceptable projects include, for example:

1. A literature survey of the research in a certain area.
2. Identifying a practical, previously unknown exploit.
3. Developing a practical, previously unknown counter-measure for an exploit.
4. Prototyping some new capability such as a circumvention technology.
5. Making a substantial contribution to some network-security relevant open source project.

Early in the term I'll ask you to send me ideas for your candidate projects so that I can approve them. This project proposal will be graded as a homework assignment. Please make use of office hours to discuss project ideas. Do not feel bound by the topics I have listed.

Any grading, exam, project or homework policies which are in violation of standard operating procedure at NJIT will be corrected to conform to NJIT policy. All times are Eastern.

8 SUGGESTIONS FOR SUCCESS

For most students this course will at times be “challenging”, and at times “easy”. However, independent, big-picture thinking is most important to your success. Please challenge yourself and your classmates to see the forest from the trees.
### 9 TENTATIVE SCHEDULE

<table>
<thead>
<tr>
<th>Date</th>
<th>Lecture #</th>
<th>Theme</th>
<th>Lecture Hour 1</th>
<th>Lecture Hour 2</th>
<th>Lecture Hour 3</th>
<th>Due</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/04/14</td>
<td>1</td>
<td>Networks</td>
<td>Introductions &amp; Policies</td>
<td>Network Stack</td>
<td>Protocol Families</td>
<td></td>
<td>First day of class</td>
</tr>
<tr>
<td>09/11/14</td>
<td>2</td>
<td>Malware</td>
<td>Common vulnerabilities</td>
<td>Malware</td>
<td>Counter-measures</td>
<td>HW Week 1</td>
<td></td>
</tr>
<tr>
<td>09/18/14</td>
<td>3</td>
<td>Measurements</td>
<td>Network Measurements</td>
<td>Tools 1</td>
<td>Tools 2</td>
<td>HW Week 2</td>
<td></td>
</tr>
<tr>
<td>09/25/14</td>
<td>4</td>
<td>Circumvention</td>
<td>Firewalls</td>
<td>Tor and friends</td>
<td>Browser Fingerprinting</td>
<td>HW Week 3</td>
<td></td>
</tr>
<tr>
<td>10/02/14</td>
<td>5</td>
<td>Wireless</td>
<td>Wireless protocols</td>
<td>WEP and WPA</td>
<td>Cellular Data Services</td>
<td>HW Week 4/Proj. Proposals</td>
<td></td>
</tr>
<tr>
<td>10/09/14</td>
<td>6</td>
<td>Web</td>
<td>DNS</td>
<td>SSL/TLS</td>
<td>Exam Topics</td>
<td>HW Week 5</td>
<td></td>
</tr>
<tr>
<td>10/16/14</td>
<td>7</td>
<td>Exam</td>
<td>Exam</td>
<td>Exam 1</td>
<td>Professor Travel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/23/14</td>
<td>8</td>
<td>Crypto</td>
<td>Encryption Standards</td>
<td>Public Key Crypto</td>
<td>AES and friends</td>
<td>HW Week 6</td>
<td>Potential Travel</td>
</tr>
<tr>
<td>10/30/14</td>
<td>9</td>
<td>Key Management</td>
<td>PKI</td>
<td>Diffie-Hellman</td>
<td>Chap / Kerberos</td>
<td>HW Week 8</td>
<td>Last day to withdraw</td>
</tr>
<tr>
<td>11/06/14</td>
<td>10</td>
<td>Embedded Systems</td>
<td>Embedded Networking</td>
<td>CAN Bus</td>
<td>Automotive Attack Surfaces</td>
<td>HW Week 9</td>
<td></td>
</tr>
<tr>
<td>11/13/14</td>
<td>11</td>
<td>TBD - selected by students</td>
<td>Chat</td>
<td>XMPP</td>
<td>OTR</td>
<td>HW Week 10</td>
<td></td>
</tr>
<tr>
<td>11/20/14</td>
<td>12</td>
<td>TBD - selected by students</td>
<td>Physical Access Control</td>
<td>Physical Security Networks</td>
<td>Biometrics</td>
<td>HW Week 11</td>
<td></td>
</tr>
<tr>
<td>11/25/14</td>
<td>13</td>
<td>Review and Discuss Projects</td>
<td>ARP</td>
<td>SOAP</td>
<td>Project Discussion</td>
<td></td>
<td>Meet Wed. 11/26/2014</td>
</tr>
<tr>
<td>12/04/14</td>
<td>14</td>
<td>Wrapping up</td>
<td>Review Course Material (as requested)</td>
<td></td>
<td></td>
<td>HW Week 12&amp;13</td>
<td>Last day of Class</td>
</tr>
<tr>
<td>TBD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Exam 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CS 683: Software Project Management
Syllabus, Fall 2015

Professor Gerard Ryan
GITC 4303
973.642.4029
gerald.ryan@njit.edu
gwryan@njit.edu
http://web.njit.edu/~gwryan
Twitter: @gwryanNJIT

Fall 2015 Office Hours: Monday 10:00-11:30, Thursday 5:00-6:00, or by appointment

<table>
<thead>
<tr>
<th>CS 683 - Software Project Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>This course gives the student the necessary background to allow her/him to manage software projects; this includes economic, managerial and organizational aspects. The essence of software engineering is not only to introduce a valuable software product, but to do so economically and competitively. Like any engineering discipline, software engineering depends critically on managerial, economic and organizational considerations. Students will learn software management technique, various software costing techniques including COCOMO and ROI, team organization and management, and various methods of software development including Cleanroom and Agile</td>
</tr>
</tbody>
</table>

Please include CS683 in the Subject: line of any email you send; it will make it easier to manage my emails. I will do the same in emails I send to you.

Course Webpage http://web.njit.edu/~gwryan/CS683
All course materials, including lecture notes, papers, etc, will be posted on the web page. Announcements and notices will also be posted. WHEN IN DOUBT, CHECK THE WEBSITE. Moodle will be used for online Q&A and for any paper or presentation submissions

Textbook

Lists of interesting additional reading will be on the course website.

Papers and articles will also be posted.
Grading

<table>
<thead>
<tr>
<th>Grading Category</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance/Participation</td>
<td>10%</td>
</tr>
<tr>
<td>Reaction Paper</td>
<td>25%</td>
</tr>
<tr>
<td>Presentation</td>
<td>30%</td>
</tr>
<tr>
<td>Final</td>
<td>35%</td>
</tr>
</tbody>
</table>

Simple rules

I don’t accept late work.  
I don’t give extra credit.  
Turn off the toys (phones especially).  
Don’t “friend” me on Facebook or any other social media.  
I expect you to behave professionally.  
Speak to me about any issues you have.

Ethical Conduct (This should go without saying, people. Seriously.)

Cheating during in-class tests or take-home examinations or homework is, of course, illegal and immoral.

The essential quality of the NJIT University Code on Academic Integrity is that each student shall demonstrate honesty and integrity in the completion of all assignments and in the participation of the learning process. Adherence to the University Code on Academic Integrity promotes the level of integrity required within the university and professional communities and assures students that their work is being judged fairly with the work of others.  
See http://www.njit.edu/academics/pdf/academic-integrity-code.pdf

Reaction Paper

Each student will be assigned a paper from a list of papers in the topic area. You will be assigned to write a “reaction paper” in which you will explain the key points in the paper and provide a critique of the topic.

Presentation

The class will be divided into 8-10 small teams. Each team will be given an assignment and will be required to make a presentation to the class on that assignment. The assignment may be to review a case study, to explain a topic area in project management, or to assess the use of various techniques in PM.

You may need to work with your project team outside of lecture hours to prepare your presentations and submissions.
Course Topics

- Introduction
- Project Planning
- Requirements
- Architectures
- Estimation
- Scheduling
- Reviews
- Management and Organization
- Change Management
- Risk Management
- Extreme Programming, Agile, Lean Startup
- What’s old and what’s new

Important Dates

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-Nov</td>
<td>Reaction Paper Due</td>
</tr>
<tr>
<td>26-Nov</td>
<td>NO CLASSES</td>
</tr>
<tr>
<td>3-Dec</td>
<td>Presentations</td>
</tr>
<tr>
<td>10-Dec</td>
<td>Presentations</td>
</tr>
<tr>
<td>17-Dec</td>
<td>Final Exam (I think)</td>
</tr>
</tbody>
</table>
CS 698: Special Topics in Big Data

General Information
Instructor: Chase Wu
Office/Lab: GITC 4107
E-mail: chase.wu@njit.edu
Phone: 973-642-4579
Course website: https://web.njit.edu/~chasewu/Courses/Fall2015/CS698BigData/CS698_BigData_Fall15.html

Department office: GITC 4400
Department phone: 973-596-5778

Course Description
This course provides an in-depth coverage of special topics in big data from data generation, storage, management, transfer, to analytics, with focuses on the state-of-the-art technologies, tools, architectures, and systems that constitute big-data computing solutions in high-performance networks. Real-life big-data applications in various domains (particularly in sciences) are introduced as use cases to illustrate the development, deployment, and testing of a wide spectrum of emerging big-data solutions.

Prerequisite
- CS 610: Data Structures and Algorithms
- CS 630: Operating System Design
- Or permission of instructor

Textbook

Resources
Additional reading materials including reference books and online resources will be assigned for some advanced topics as the course proceeds.

Evaluation
Grading components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>10%</td>
</tr>
<tr>
<td>Homework</td>
<td>10%</td>
</tr>
<tr>
<td>Project</td>
<td>20%</td>
</tr>
<tr>
<td>Midterm</td>
<td>30%</td>
</tr>
<tr>
<td>Final</td>
<td>30%</td>
</tr>
</tbody>
</table>

Grading scale*:

<table>
<thead>
<tr>
<th>Grade</th>
<th>CS 698</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90 - 100</td>
</tr>
<tr>
<td>B</td>
<td>80 - 89</td>
</tr>
<tr>
<td>C</td>
<td>70 - 79</td>
</tr>
<tr>
<td>D</td>
<td>60 - 69</td>
</tr>
<tr>
<td>F</td>
<td>59 and below</td>
</tr>
</tbody>
</table>

*Final grades will not be curved unless necessary.

Late Policy
Students are expected to complete work on schedule. Late work is not accepted unless prior arrangements are made with the instructor.
Academic Integrity and Student Conduct:

*Plagiarism or cheating* behavior in any form is unethical and detrimental to proper education and will not be tolerated. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student’s own work. The plagiarism is incurred when any part of anybody else’s work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but appropriate references must be included for the materials consulted, and appropriate citations made when the material is taken verbatim.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor’s discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the Dean of Students for further disciplinary action. For further information on the Code of Student Conduct and academic discipline procedures, please refer to: [http://www.njit.edu/doss/policies/conductcode](http://www.njit.edu/doss/policies/conductcode).

**Course Syllabus**

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>• Introduction</td>
</tr>
<tr>
<td>2</td>
<td>• In-class Presentation</td>
</tr>
<tr>
<td>3</td>
<td>• Trends of Computing for Big Data</td>
</tr>
<tr>
<td></td>
<td>o High-performance Computing (Supercomputers and Clusters)</td>
</tr>
<tr>
<td></td>
<td>o Grid Computing</td>
</tr>
<tr>
<td></td>
<td>o Cloud Computing</td>
</tr>
<tr>
<td></td>
<td>o Mobile Computing</td>
</tr>
<tr>
<td>4, 5</td>
<td>• Big Data Overview</td>
</tr>
<tr>
<td></td>
<td>o Drivers of Big Data</td>
</tr>
<tr>
<td></td>
<td>o Big Data Attributes</td>
</tr>
<tr>
<td></td>
<td>o Data Structures</td>
</tr>
<tr>
<td></td>
<td>o Big Data Ecosystem</td>
</tr>
<tr>
<td></td>
<td>o Examples of Data Analytics</td>
</tr>
<tr>
<td>6, 7</td>
<td>• Big Data Tools, Techniques, and Systems</td>
</tr>
<tr>
<td></td>
<td>o Exascale Computing</td>
</tr>
<tr>
<td></td>
<td>o HDFS, HBase, and NoSQL</td>
</tr>
<tr>
<td></td>
<td>o MapReduce</td>
</tr>
<tr>
<td></td>
<td>o Hadoop 1 and Hadoop 2 (YARN)</td>
</tr>
<tr>
<td>8, 9</td>
<td>• Advanced Analytical Theory and Methods</td>
</tr>
<tr>
<td></td>
<td>o Clustering</td>
</tr>
<tr>
<td></td>
<td>o Classification</td>
</tr>
<tr>
<td></td>
<td>o Regression</td>
</tr>
<tr>
<td></td>
<td>o Visualization</td>
</tr>
<tr>
<td>10, 11</td>
<td>• Review and Midterm Exam</td>
</tr>
<tr>
<td>12, 13, 14</td>
<td>• Advanced Topics</td>
</tr>
<tr>
<td></td>
<td>o High-performance Networking for Big Data Movement</td>
</tr>
<tr>
<td></td>
<td>o Scientific Workflow Management and Optimization</td>
</tr>
<tr>
<td>15</td>
<td>• Review</td>
</tr>
</tbody>
</table>
CS 704 --- Sequencing and Scheduling
IE 704 --- Sequencing and Scheduling

Weekly Listing of Course Topics

Week 1
  Overview of Scheduling. The 3-field Notation. Tutorial on Complexity Theory.

Week 2
  Tutorial on Complexity Theory.

Week 3

Week 4
  Muntz-Coffman Algorithm.

Week 5
  List Scheduling. Anomaly of List Scheduling. Comparison of List, Nonpreemptive and
  Preemptive Scheduling.

Week 6

Week 7
  Midterm Exam.

Week 8
  Mean Weighted Flow Time. SPT Rule. Bicriteria Scheduling: (1) Minimizing
  Makespan Subject to Minimum Mean Flow Time.

Week 9
  Bicriteria Scheduling (2) Minimizing Mean Flow Time Subject to Minimum
  Makespan. Minimizing Mean Flow Time Subject to Release Time and Deadline
  Constraint.

Week 10
  Maximum Lateness Objective. Lawler’s Algorithm. Sahni’s Algorithm. Horn’s
  Network Flow Algorithm.

Week 11
  Hodgson-Moore Algorithm. Tardiness Objective. Lawler’s Pseudo-Polynomial Time
  Algorithm.
Week 12
Midterm Exam.

Week 13
Open Shop, Flow Shop and Job Shop

Week 14
Review