BNFO 135: Programming For Bioinformatics

Syllabus

Instructor Info

Instructor: Jonathan Kapleau
Office: GITC 4412
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

<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>Homeworks</td>
<td>15%</td>
</tr>
<tr>
<td>------------</td>
<td>-----</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**

None
CS 100 Roadmap to Computing
Course Syllabus, Spring 2016

Course Description
An introduction to programming and problem solving skills using Python, a very high level language. Topics include basic strategies for problem solving, conditional, repetition, function and other 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 includes a project in which the student investigates and reports on a topic of current interest in computing.

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, the student, in addition to mastering the programming and problem solving materials, is expected to learn to effectively use learning strategies and materials – learning how to learn efficiently in preparation for a knowledge intensive profession. This includes effective use of knowledge resources -- reading documentation, asking and answering peer questions, consulting with more experienced persons, and searching on-line for answers. It also includes tools and methodology – testing to verify the correctness of code, use of an IDE and debugger, writing specifications and documentation.

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

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 pythontutor.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. This is an important checkpoint in assuring your grasp of the material being covered and correctly solving assigned problems. When you go for recitation you should have already read the assigned material and worked on current homework. You should be prepared during recitation to check whether your understanding of this material is correct, to pose and answer questions and to 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 when 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 a specified problem:

- 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, February 8 and Monday, March 21, 4:00-5:45pm.
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 common 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 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 101-102 Spring 2016: Computer Program & Problem Solving
Thursday 6:00PM~9:05PM GITC2315C

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

Teaching Assistant: TBA

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 33% 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**
1. All your work, including homework assignments and the project, must be submitted electronically via the course Moodle page. No email submission will be accepted.
2. Copying programs or written assignments (from any sources, no matter your classmates or internet) is a violation of NJIT honor code. You must come up with your own solutions and all written work must be your own. If you want to discuss homework questions or the project with your classmates, use only general terms.
3. 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.
4. 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.
5. 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.
6. Turn off all mobile devices (e.g. cell phones and laptops) during class unless you are otherwise instructed.
7. Visit the course homepage on Moodle regularly and frequently for lecture notes, homework assignments, instructions, and latest updates.

**Honor Code**

Students must follow The NJIT Honor Code. Any violations will be brought to the immediate attention of the Dean of Students.
CS 103: Computer Science With Business Problems

Syllabus

Instructor Info

Instructor: Jonathan Kapleau
Office: GITC 4412
Office Phone: 973-596-2657

Course Description

An introductory course in computer science, with applications to business and managerial decision making. Topics include basic concepts of computer systems, software engineering, algorithm design, programming languages and abstraction, with applications. The Java programming language will be used to illustrate concepts and techniques.

Textbooks

Java Software Solutions
John Lewis, William Loftus

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>20%</td>
</tr>
<tr>
<td>Homework</td>
<td>10%</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10%</td>
</tr>
</tbody>
</table>

Topics

- Introduction to Java
- Variables and Expressions
- Using Classes
• Creating Classes
• Control Structures
• Arrays

**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**

None
CS104: Computer Programming and Graphics Problems
Spring 2016

Course Identification:

Course: Computer Programming and Graphics Problems
Instructor: Osama Eljabiri, PhD
Office: Room 4210 - GITC Building – 4th Floor
Office Hours: Thursdays 4-5:30 PM
  Walk-ins are welcome any time based on availability
  Online and in-class help hours are also available (see below)
Class-based extended office hours: 15-30 Minutes before most classes and up to one hour after most evening classes (as necessary).
Live office hours: Online assistance and orientation available via email
Telephone  (973) 642-7123 or Cell Phone: (732) 456-0249 (preferred)
FAX: (866) 605-9416
E-mail: oe2@njit.edu or eljabiri@gmail.com
Class Time/Location: As posted on the registrar system and MOODLE
Virtual Classroom System: http://MOODLE.njit.edu (or moodle.njit.edu)

1. Course Information:

A. Course Number, Title, Credits
CS104, Computer Programming and Graphics Problems, 3 credits.

B. Prerequisites
Corequisite: Math 138

C. Catalogue Course Description

An introductory course in computer science with applications in computer graphics for architecture. Emphasis on programming methodology using a high level language as the vehicle to illustrate the concepts. Topics include basic concepts of computer systems, software engineering, algorithm design, programming languages and data abstraction, with applications.

2. Course Features and Objectives:
A- Features:
This course has unique features that are not currently offered through any other course on campus. These features are:
- It provides hands-on multidisciplinary real world experiences that integrate business applications with computer technology areas such as art & design, multimedia and game development.
- It simulates the real-world environment internally in the structure of students’ teams and course “virtual organization”.
- It offers dynamic market-driven training that reflects hot topics highly demanded by industry but not usually covered through a static college curriculum.
- It enables students to master career-oriented skills such as leadership, presentation, entrepreneurship, social and communication skills.
- It shows how both IT and business knowledge are used to solve real-world architecture-related problems.
- The experience gained working on such projects will make students more employable by industry including the ability of building businesses through the entrepreneurship track.

B- Specific goals for the course
Students who complete this course successfully will have:

- Ability to breakdown complex problems into manageable pieces (using WBS and Gantt).
- Ability to define project stakeholders, scope & requirements (including the use of FDD).
- Ability to capture, map and visualize the design of the proposed solution identifying key components and their relationships.
- Ability to implement the solution successfully using software and/or hardware technologies with emphasis on Database design and development.
- Ability to communicate a value proposition of the project to various stakeholders including the ability to explain, convince, engage and impress.
- Ability to organize the presentation in a meaningful and professional fashion including mastering personal and collaboration presentation skills.

Accordingly, the general outcomes of this course include:
(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
(d) An ability to function effectively on teams to accomplish a common goal
(e) An understanding of professional, ethical, legal, security and social issues and responsibilities
(f) An ability to communicate effectively with a range of audiences
(h) Recognition of the need for and an ability to engage in continuing professional development (i) An ability to use current techniques, skills, and tools necessary for computing practice.
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.
3. Course Details:

A. Course outline with approximate week-by-week schedule (subject to change if necessary)

CS104 Course Schedule – Spring 2016 – attached in the excel spreadsheet

B. References
No textbook is required.

C. Number of hours of lecture, recitation, and laboratory

Students work on real-world entrepreneurial, industry sponsored or research development projects for the entire 14 weeks of the semester. The problem solving process is broken down into five “Sprints”. Lectures and training will include a comprehensive crash course on weekly basis and some on-demand training throughout the semester. Additional hands-on training, project management training and laboratory hours will also be included.

GRADING POLICIES

Your final grade in this course will be based on the percentage of points that you receive out of the total possible points for the course (1100). Grades will be determined according to the following scale:

90% - 100%    A
85% - 89%      B+
80% - 84%      B
75% - 79%      C+
70% - 74%      C
60% - 69%      D
0% - 59%       F
# CS104 Evaluation Criteria
## Spring 2016

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Points/ Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>200 Points (20%)</td>
</tr>
<tr>
<td>Weekly Activities and Participation</td>
<td>200 points (18%)</td>
</tr>
<tr>
<td>Midterm Assessment</td>
<td>200 points (20%)</td>
</tr>
<tr>
<td>Database Experience and Sub-Project</td>
<td>150 points (15%)</td>
</tr>
<tr>
<td><strong>Team Project</strong> <em>(presentation, report and final product)</em></td>
<td><strong>250 Points (25%)</strong></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1000 points (100%)</strong></td>
</tr>
</tbody>
</table>

- You can earn up to 150 points extra credit (15%).

- Curve is possible if necessary. Please don't count on it since it might not happen at all.

- Wights are subject to change if necessary.

Please note that:

- Class attendance, and in-Class /online participation and collaboration is very important.
- In-group participation and attendance is extremely significant in determining your final letter grade.

Good Luck,
Osama Eljabiri
CS 113 - Introduction to Computer Science I  
Course Syllabus Spring 2016

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: GITC 1100  
Instructor: Dr. Michael A. Baltrush

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

- Recitation Section 002: Monday: 8:30-9:55 AM  
  Room: GITC 2315A

- Recitation Section 004: Monday: 2:30-4:00 PM  
  Room: GITC 2315C

- Recitation Section 006: Monday: 8:30 am – 9:55 am  
  Room: GITC 2315C

Contact Information

Instructor: Dr. Michael A. Baltrush
Office: GITC 4310
Email: baltrush@njit.edu
Office Hours:  
  Tuesday 2:30-3:10 PM  
  Thursday 2:30-3:10 PM  
  Tuesday 4:00-4:40 PM  
  By appointment

TA Contact Information

TA: Xiaoyuan Liang: Sections 002/004
Office: GITC 4325
Email: xl367@njit.edu
Office Hours: T: 1:30-2:30, R 1:30-2:30
TA: J. Spirollari: Section 006
Office: GITC 4405
Email: js9@njit.edu
Office Hours: M: 12:00-12:55, W: 1:30-2:30
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. Neither graphics nor applets are part of this course.

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 NJITs lab classrooms.

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

www.jgrasp.org

You will also need to install 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.
Prerequisite

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 via the Dean of Students office.
- All submitted work (including exams) must include your name and UCID.
- 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 have accomplished the following:

1. Learned how to use core Java facilities with a focus on problem solving
2. Learned 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 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. Your understanding of programming concepts.

Evaluation

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

- Homework 33%
- Midterm Exam 34%
- 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. The 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>3-4</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></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>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>---</td>
<td>-------------------------------------------------------</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></th>
<th></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>Assigned in Lecture, due in recitation listed Hand in and pick up graded assignment in recitations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>January 21, 2016</td>
</tr>
<tr>
<td>2</td>
<td>February 4, 2016</td>
</tr>
<tr>
<td>3</td>
<td>February 18, 2016</td>
</tr>
<tr>
<td>4</td>
<td>March 3, 2016</td>
</tr>
<tr>
<td>5</td>
<td>March 24, 2016</td>
</tr>
<tr>
<td>6</td>
<td>April 7, 2016</td>
</tr>
<tr>
<td>7</td>
<td>April 21, 2016</td>
</tr>
</tbody>
</table>
Readings From

Java Software Solutions by Lewis & Loftus – 8/E

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
Overview
This course is a comprehensive introduction to the Java programming language teaching writing, testing and debugging of programs. The course has three major parts. The first part teaches fundamental programming techniques that use primitive data types, variables, assignments expressions and operators, control statements, arrays and files I/O. The second part covers testing and debugging and teaches students how to write programs that work reliably. The third part introduces object-oriented programming. The course guides students to the development of comprehensive Java applications.

Textbook

Prerequisites
CS100 – Roadmap to Computing or equivalent

Course Policies
Attendance is mandatory. A student who misses more than 5 classes will be dropped, without credit.

You must bring the textbook to class.

Moodle (http://moodle.njit.edu/) will be used for course discussion.

Homework must be submitted via Moodle 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 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 covered

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

- Recursion  
- Exceptions

**Learning Outcomes**

Upon completing the course, the students will be expected to know and be able to use these elements to compute the solution to a problem:

- Understand the concept of classes and objects
- Design and implement own classes
- Create and use correctly object of different types
- Devise a sequence of steps (algorithm) that correctly solves a given problem.
- Write a program that implements the algorithm using:
  - A main set of java programming language elements (variables, syntax, keywords)
  - Data types (primitive and object data types including arrays)
  - Statements that perform input/output, control statements
  - Exception handling
- Understand inheritance and polymorphism and correctly use to solve complex problems
- Understand recursion and implement recursive methods

**Evaluation**

The evaluation will be based on the following course requirements:

- Homework 30%
- Midterm Exam 31%
- Final Exam 33%
- Misc. 6%

**Exam Policies**

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

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 other midterm.

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.
University Code on Academic Integrity

Every student should read the University Code on Academic Integrity (http://www.njit.edu/academics/integrity.php). All work that you represent as your own must, in fact, 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 and the corresponding reading from the suggested textbook.

<table>
<thead>
<tr>
<th>Week (Approx)</th>
<th>Topic</th>
<th>Reading from book</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to programming and Java programming language</td>
<td>Ch. 1</td>
</tr>
<tr>
<td>2</td>
<td>Data and Expressions</td>
<td>Ch. 2</td>
</tr>
<tr>
<td>2,3</td>
<td>Using Classes &amp; Objects,</td>
<td>Ch. 3</td>
</tr>
<tr>
<td>3</td>
<td>Writing Classes</td>
<td>Ch. 4</td>
</tr>
<tr>
<td>4,5</td>
<td>Writing Classes</td>
<td>Ch. 4</td>
</tr>
<tr>
<td>5,6</td>
<td>Conditional &amp; Loops</td>
<td>Ch. 5, 6</td>
</tr>
<tr>
<td>7,8</td>
<td>Arrays</td>
<td>Ch. 8</td>
</tr>
<tr>
<td>9</td>
<td>Object-Oriented Design (Testing &amp; Debugging)</td>
<td>Ch. 7</td>
</tr>
<tr>
<td>10,11</td>
<td>Inheritance</td>
<td>Ch. 9</td>
</tr>
<tr>
<td>12</td>
<td>Polymorphism</td>
<td>Ch. 10</td>
</tr>
<tr>
<td>13</td>
<td>Recursion</td>
<td>Ch. 12</td>
</tr>
<tr>
<td>14</td>
<td>Exception Handling</td>
<td>Ch. 11</td>
</tr>
</tbody>
</table>
CS 113H: Honors Introduction To Computer Science I

Syllabus

Instructor Info

Instructor:  Jonathan Kapleau
Office:      GITC 4412
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.

Textbooks

Java Software Solutions
John Lewis, William Loftus

Grading Scheme

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Final</td>
<td>25%</td>
</tr>
<tr>
<td>Projects</td>
<td>40%</td>
</tr>
<tr>
<td>Homeworks</td>
<td>5%</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10%</td>
</tr>
</tbody>
</table>

Topics

- Data And Expressions
- Classes And Objects
- Selection Structures
• Iterative Structures
• Object Oriented Design
• Arrays
• Encapsulation
• Inheritance
• Polymorphism
• Exception Handling
• Recursion
• Collections

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. subsequent semester.

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

CS 100 or equivalent.
CS 115: Introduction To Computer Science I in C++

Syllabus

Instructor Info

Instructor: Jonathan Kapleau
Office: GITC 4412
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. C++ is used in this course.

Textbooks

![Absolute C++](image)

Absolute C++
Walter Savitch
ISBN: 0136083811

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>20%</td>
</tr>
<tr>
<td>Homeworks</td>
<td>10%</td>
</tr>
<tr>
<td>Labs</td>
<td>10%</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>10%</td>
</tr>
</tbody>
</table>

Topics

- Introduction to C++
- Flow of Control
- Function Basics
- Parameters and Overloading
- Arrays
- Structures and Classes
- Constructors
- Operator Overloading and Friend Functions
- Recursion
- Inheritance
- Polymorphism
- Exception Handling

**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**

None
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 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:
1. CS 114: Intro to Computer Science;
2. Math 112: Calculus II.


Course Objectives (what you are expected to get out of this course)
1. Learn basic mathematical tools and terminologies used in computer science
2. Learn propositional logic, reasoning, and basic proof techniques
3. Learn induction, recursion, recurrence equations, and divide-and-conquer technique
4. Learn mathematical tools to analyze efficiency of algorithms
5. Implement some simple programs and run experiments to measure their running time
6. Learn permutations/combinations, basic discrete probability, and applications

Course Evaluation (Assessment):

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Exam Dates:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments 25%</td>
<td></td>
</tr>
<tr>
<td>Exam 1 25%</td>
<td>Week 6, Thurs Feb 25</td>
</tr>
<tr>
<td>Exam 2 25%</td>
<td>Week 11, Thurs April 7</td>
</tr>
<tr>
<td>Final Exam 25%</td>
<td>Finals week Date to be announced by Registrar</td>
</tr>
</tbody>
</table>

Notes: Pictured NJIT ID required for all exams. All exams are closed books/notes.

Policies:
1. Assignments must be done by you individually. Team-work not allowed.
2. Submit paper-copy of assignments at the start of the class period on the due date.
3. Late assignments will not be accepted.
4. Website: You must check the course website regularly for posting of syllabus, assignments, announcements, and old exams.
5. **Academic Integrity**: Familiarize yourself with NJIT Honor Code: [http://integrity.njit.edu](http://integrity.njit.edu). Any evidence of dishonesty will be dealt with seriously and reported to the Dean of Students.

### CS 241 Course Outline

<table>
<thead>
<tr>
<th>Week (Approx)</th>
<th>Topic</th>
<th>Reading from Johnsonbaugh (7th Ed.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Algebra Self-Review</strong></td>
<td>Appendix B, pp.631-643</td>
</tr>
<tr>
<td>1-2</td>
<td><strong>Sets and Logic</strong></td>
<td>Ch. 1</td>
</tr>
<tr>
<td></td>
<td>Sets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Propositional Logic (Application: Google Search)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quantifiers (Application: Database operations)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Proof Techniques</strong></td>
<td>Ch. 2</td>
</tr>
<tr>
<td></td>
<td>Direct Proof, Counterexample, Contrapositive, Proof by Contradiction,</td>
<td>Skip 2.3: Resolution proofs</td>
</tr>
<tr>
<td></td>
<td>Proof by Induction; Strong Induction</td>
<td></td>
</tr>
<tr>
<td>4-5</td>
<td><strong>Functions and Relations</strong></td>
<td>Ch. 3</td>
</tr>
<tr>
<td></td>
<td>Functions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Properties: Reflexive, Symmetric, Transitive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Partial Order, Total Order, Equivalence Relations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Matrices of Relations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Application: Relational Databases</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>Exam 1 (See page 1 for the date)</strong></td>
<td></td>
</tr>
<tr>
<td>7-8</td>
<td><strong>Algorithms</strong></td>
<td>Ch. 4</td>
</tr>
<tr>
<td></td>
<td>Analysis of Algorithms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Recursive Algorithms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use of Recurrences to Analyze Algorithms</td>
<td></td>
</tr>
<tr>
<td>9-10</td>
<td><strong>Recurrence Equations</strong></td>
<td>Ch. 7</td>
</tr>
<tr>
<td></td>
<td>Divide-and-Conquer Recurrences</td>
<td>Posted Notes</td>
</tr>
<tr>
<td></td>
<td>Master Theorem</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linear Recurrences</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><strong>Exam 2 (See page 1 for the date)</strong></td>
<td></td>
</tr>
<tr>
<td>12-13</td>
<td><strong>Counting Methods</strong></td>
<td>Ch. 6</td>
</tr>
<tr>
<td></td>
<td>Permutations and Combinations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Principle of Inclusion/Exclusion</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pigeonhole Principle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduction to Basic Probability</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><strong>Introduction to Number Theory</strong> (if time permits)</td>
<td>Ch. 5</td>
</tr>
<tr>
<td>15</td>
<td><strong>Introduction to Trees and Graphs</strong> (if time permits)</td>
<td>Ch. 8, 9</td>
</tr>
</tbody>
</table>
CS252 Computer Organization and Architecture
Spring 2016
Instructor: Dr. Michael A. Baltrush

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


Class: CS252-002 TR 1:00-2:30 PM, CKB 219
       CS252-102 T 6:00-9:00 PM CKB 212

Office Hours: Tuesday 2:30-3:10 PM
              Thursday 2:30-3:10 PM
              Tuesday 4:00-4:40 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: February 23, 2016
   Exam2: April 12, 2016
1/3 Final exam is cumulative: Scheduled by Registrar; day, May 10, 2016: evening
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 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>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1</td>
<td>Basic Structure of Computers</td>
<td>Two’s Complement representation of integers Floating Point</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Machine Instructions and programs</td>
<td>Operand Addressing</td>
</tr>
<tr>
<td>Appendix D</td>
<td>ARM Instruction Set and Examples</td>
<td>Machine Instructions</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Input/Output Organization</td>
<td>Subroutine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARM Instruction set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARM IDE</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Basic Processing Unit</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 6</td>
<td>Pipelining</td>
<td>Instruction execution</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Input/output Organization</td>
<td>Hardware Components</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control signals</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hardwired/Microprogram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Control</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Memory system</td>
<td>Basics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Various Hazards</td>
</tr>
<tr>
<td>Chapter 9</td>
<td>Arithmetic</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 10</td>
<td>Coding/Compression</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></td>
<td></td>
<td>Integer Adders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multiply Integers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Divide Integers</td>
</tr>
<tr>
<td>Class Notes</td>
<td>Coding/Compression</td>
<td>Compression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Error Correcting</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.

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

Instructor Info

Instructor: Jonathan Kapleau
Office: GITC 4412
Office Phone: 973-596-2657

Course Description

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.

Textbooks

Concepts of Programming Languages, 11th edition
Robert Sebesta
ISBN: 013394302X

Grading Scheme

<table>
<thead>
<tr>
<th></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

- Common Features of Programming Languages
- Lexical Syntax
- Grammars
- Names
- Types
- Semantics
- Expressions
- Control Flow
- Subprograms
- Memory Management
- Event Handling
- Concurrency

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

None
CS 280: Programming Language Concepts
Syllabus, Spring 2016

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

Fall 2016 Office Hours: Tuesday 1:00-2:30, Friday 2:30-4:00, 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 be posted on http://web.njit.edu/~gwryan
and sent by email.

Textbook:

Grading

<table>
<thead>
<tr>
<th>Programs</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>27%</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

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 without attribution. You may NOT use code in your assignments that you did not write without attribution. Giving your work to someone else to copy from is just as much cheating as copying someone else’s work.

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 assignments will be in C++. You will NOT need to submit a printout of your code.

- 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

- Every file 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 #

- 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>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10-Feb</td>
<td>Program 1 Due</td>
</tr>
<tr>
<td>7-Mar</td>
<td>Program 2 Due</td>
</tr>
<tr>
<td>7-Mar / 9-Mar</td>
<td>Midterm</td>
</tr>
<tr>
<td>11-Apr</td>
<td>Program 3 Due</td>
</tr>
<tr>
<td>3-May</td>
<td>Program 4 Due</td>
</tr>
<tr>
<td>6-May - 12-May</td>
<td>FINALS WEEK</td>
</tr>
</tbody>
</table>
Spring 2016 CS288 Intensive Programming in Linux - Syllabus

- 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: MW 11:30am-1pm. You can come anytime during the office hours. No need to tell me you are coming.
- Teaching assistant: no TA, NJIT policy
- Grader: TBA if get one
- Class time and location: MW 10 - 11:25 am GITC 1400, See the registrar's page http://www.njit.edu/registrarschedules/courses/spring/2016S.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, Mon, 2/22/2016 (30%); Test 2, Mon, 4/4/2016 (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. You'll be glad you did down the road.
- 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 University policy.
- Exam related. Please peruse:
  - Exam papers will not be returned to you. A copy of the first page showing your score will be returned to you. The exam papers are kept for accreditation purposes. You may come and view your graded exam paper in my office during the office hours.
  - Procedure for exams: An hour or two before the exam, you will receive an email regarding seating assignment based on NJIT ID number. Make sure to take a seat according to the seating assignment. Numbered exams will be distributed to the matching seats. Make sure your numbered exam paper matches your seat assignment. Leave your NJIT ID card on the table. IDs will be checked during the exam. Classroom pictures will be taken to record those present to take exam Do not take someone else's exam paper. You both are taking someone else's exam.
  - 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.
  - 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 on 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.
- See Academic Integrity
Spring 2016 CS288 Intensive Programming in Linux - Syllabus

Lecture schedule - Contents may change according to the 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. Test 1, Mon, 2/22/2016
11. STAGE 2 - tools for building an end-to-end realworld application
    Sorting - fast integer radix sort for integers
12. Sorting - introduction to floating point representation
13. Sorting - fast radix sort for floats
14. State space search - depth first, breadth first search
15. State space search - heuristic-based intelligent search
16. State space search - intelligent bi-directional search
17. Matrix computation: a system of linear equation solvers
18. Matrix computation: introduction to iterative methods
19. Matrix computation: application to spectral graph partitioning
21. STAGE 3 - an end-to-end realworld application
    Web processing - fetching with wget using Bash scripting, intro to DOM tree, properties, methods
22. Web processing - DOM tree navigation, data extraction using Python minidom
23. Web processing - getting up and running mySql server, mySql DB construction, data injection
24. Web processing - getting up and running Apache server, reading DB using PHP, constructing clickable sortable, plotting charts and graphs and presenting data using PHP
25. STAGE 4 - extending tools and applications to run on many-core machines
26. Introduction to multicore/parallel computing using MPI (optional OpenMP) - point to point communication
27. Introduction to MPI (optional OpenMP) - collective group communication
28. Simple matrix computation for multicore/multiple machines using MPI (optional parallel radix sort using MPI on a cluster of machines)
29. Final exam, see the registrar's page for the time and location
CS 341: Foundations of Computer Science II  
eLearning Section Syllabus, Spring 2016

Course Info

Instructor: Prof. Marvin K. Nakayama  
Office: GITC 4312  
Phone: 973-596-3398  
E-mail: marvin@njit.edu (preferred over phone)  
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, Thursday, 1:00pm – 2:00pm, 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:

- HW EL 10%
- Projects (3) 30%
- Midterm (1) 30%
- Final Exam 30%

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.

Students caught violating the NJIT University Code of Academic Integrity on a project will get a 0 for the project and have their course grades lowered by two steps (e.g., B+ to C+, or C to F) for each violation.

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 or violated the NJIT University Code of Academic Integrity. For each project for which you did not turn in a minimally working program, your preliminary grade will be lowered by one step. For each violation of the NJIT University Code of Academic Integrity, your preliminary grade will be lowered by two steps. For example, if your preliminary grade was B and you did not turn in a minimally working program for exactly one project, then your course grade is C+; if you did not turn in minimally working programs for exactly two projects, then your course grade drops to a C; if you did not turn in minimally working programs for three projects, then your course grade drops to a D. If you turned in minimally working programs for all projects and did not violate the NJIT University Code of Academic Integrity on any of them, 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, March 5, 2016. The final exam will be on Saturday, May 7, 2016. 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 a 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 as follows:
<table>
<thead>
<tr>
<th>Lateness (Hours)</th>
<th>Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 &lt; Lateness ≤ 24</td>
<td>10</td>
</tr>
<tr>
<td>24 &lt; Lateness ≤ 48</td>
<td>30</td>
</tr>
<tr>
<td>48 &lt; Lateness ≤ 72</td>
<td>60</td>
</tr>
<tr>
<td>72 &lt; Lateness</td>
<td>100</td>
</tr>
</tbody>
</table>

For example, since Project 1 is due by 1:00pm NJ local time on 2/16/2016, 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 10 points of the project. If you turn in a project between 24 hours and 48 hours late, then you will automatically lose 30 points of the project. If you turn in a project between 48 hours and 72 hours late, then you will automatically lose 60 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.

Students may be called in to explain their projects in person. If you are asked to come to explain your project in person, then you must do it; otherwise, you will receive a 0 on the project, and have your course grade lowered by one step.

**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 EL, 1</td>
<td>1/26</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>2/2</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>2/9</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>2/16</td>
</tr>
<tr>
<td>5</td>
<td>CFG, PDA</td>
<td>2b to 2g</td>
<td>Chapter 2</td>
<td>HW 5</td>
<td>2/23</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>3/1</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>3/8</td>
</tr>
<tr>
<td>8</td>
<td>Decidability</td>
<td>4a to 4e</td>
<td>Chapter 4</td>
<td>HW 8</td>
<td>3/22</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>3/29</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>4/5</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>4/12</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>4/19</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>4/26</td>
</tr>
<tr>
<td>14</td>
<td>Review</td>
<td>Review 1 to 6</td>
<td></td>
<td></td>
<td>5/3</td>
</tr>
</tbody>
</table>

**Important Dates**

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

- 1/26/2016, 11:55pm NJ local time: HW EL due
- 2/16/2016, 1:00pm NJ local time: Project 1 due
- Saturday, March 5, 2016: Midterm (tentatively 9:30am - noon)
- 3/22/2016, 1:00pm NJ local time: Project 2 due
- 4/19/2016, 1:00pm NJ local time: Project 3 due
- Saturday, May 7, 2016: Final Exam (tentatively 9:30am - noon)
New Jersey Institute of Technology  
Department of Computer Science

CS370 - Introduction to Artificial Intelligence -  
Spring'2016

Monday 2:30 - 5:25PM, KUPF 203

Course Description | Goals | Outcomes | Readings | Tentative Contents | Grading Policy |  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Miscellaneous</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chengjun Liu, Ph.D.

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

Office Hours: Monday 1:25-2:25PM, Tuesday 3:30-5:30PM, or by appointment

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, Search, Knowledge and Reasoning, Logical Agents, First-Order Logic and Inference, Uncertain Knowledge and Reasoning, Quantifying Uncertainty, Probabilistic Reasoning, Perception, Pictorial Knowledge Representation, and Search in Frequency and Spatial Domains. Additional topics include Machine Learning, Neural Computation, Evolutionary Computation, and Robotics.

- Prerequisites: CS 114 and (Math 226 or CS 241)

Specific Goals for the Course

- Students are prepared to work on AI related projects, such as problem solving, knowledge representation and reasoning, perception, and search.
- Students learn the introductory concepts and methodologies for artificial intelligence.
- Students learn the fundamental materials with a broad scope of applications.

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, Search, Knowledge and Reasoning, Logical Agents, First-Order Logic and Inference, Perception, Pictorial Knowledge Representation, and Search in Frequency and Spatial Domains.
Readings

- Selected papers and handouts.

Tentative Contents

1. Introduction
   - AI Fundamentals: Knowledge & Search, Cognitive Science, Turing Test, Ancient Philosophers (Logic)
   - Programming Languages: Lisp, Prolog, C/C++, Java, Matlab
   - Related Fields: Machine Learning, Neural Networks, Evolutionary Computation, Computer Vision

2. AI Basic Concepts
   - Neural Networks, Connectionism
   - Expert Systems, Symbolism
   - AI winters
   - Agents, Acting/Thinking Humanly/Rationally

3. Problem Solving
   - Intelligent Agent: Sensors, Actuators, Agent Program
   - Rational Agents: Vacuum-cleaner Agent
   - PEAS
   - Solving Problems by Searching: problem-solving agent

4. Search
   - Blind Search Strategies vs. Informed Search Strategies
   - Breadth-first Search, Depth-first Search
   - Greedy Best-first Search
   - A* Search (completeness, optimality, complexity)

5. Knowledge and Reasoning - Propositional Logic and Logical Agents
   - Knowledge Base, Models, and Knowledge-Based Agents
   - Propositional Logic Knowledge Representation Language
   - Syntax and Semantics

6. Knowledge and Reasoning - Logical Reasoning
   - Logical Reasoning: Entailment and Inference (soundness, completeness)
   - Propositional Theorem Proving: Validity, Satisfiability, Reduction to Absurd
   - MP Inference Rule, Resolution Inference Rule, Horn Form, CNF

7. Knowledge and Reasoning - First-Order Logic
   - Propositional Logic vs. First-Order Logic: objects, relations (unary, n-ary), functions
   - First-Order Logic: Syntax and Semantics (predicates, variables, quantifiers)
   - First-Order Logic Knowledge Representation Language, Model, Interpretation

8. Knowledge and Reasoning - Inference in First-Order Logic
   - Universal Instantiation, Existential Instantiation
   - Substitution and Unification
   - Generalized MP Rule, Soundness of GMP
   - Resolution Inference Rule, CNF

9. Perception - Pictorial Knowledge Representation
   - Digital Image Fundamentals
   - Image Formation
   - Digital Image Formats/Protocols (JPEG, PNG, TIFF, PGM, PPM)
   - Digital Video Fundamentals (CAV; NTSC/PAL/SECAM; S-Video)
10. Perception - Search in Frequency Domain
   - FT/FFT
   - Lowpass and Highpass Filtering
   - Convolution, Correlation, and Autocorrelation Theorems
   - Pictorial Information Search using FFT Features

11. Perception - Search in Spatial Domain
    - Geometric Feature Representation
    - Edge Detection (Canny, Zero-crossing, LOG, Prewitt, etc.)
    - Line and Curve Detection (Hough Transform)
    - Pictorial Information Search using Geometric Features

12. Learning
    - Inductive Learning
    - Decision Tree Learning
    - Neural Networks Learning

13. Evolutionary Computation (optional)
    - Genetic Algorithms (GA)
    - Evolutionary Strategy (ES)
    - Evolutionary Programming (EP)

14. Robotics (optional)
    - 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:
- MATLAB
Fundamentals of Network Security

Course No.
CS 357

Sections
102

Title
Fundamentals of Network Security

Course Websites
http://www.cs.njit.edu/~karvelas/CS-357-Spr16

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
Wednesdays: 4.30 pm - 5.50 pm
Thursdays: 5.10 pm - 5.50 pm
Fridays: 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 Quizzes: 10% each
Two lab assignments: 10% each
Midterm: 28%
Final: 32%
Topics
Introduction to Network Security
Overview of Internet Protocols
Target Reconnaissance
Target Scanning and Vulnerability Analysis
Gaining Access Techniques
Maintaining Access Techniques
Network Attacks
Denial of Service Attacks (DoS) and Distributed DoS (DDoS)
Designing an Effective Defense

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

Time & Place
CS-357-102: Wednesdays, 6.0 PM, FMH-110

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.
CS 431 Introduction to Databases
Course Syllabus + NJIT Database Access Information
Spring 2016

Section 431-002: Tuesday, Friday: 4:00 pm – 5:25 pm, Room: GITC 2400
Section 431-004: Tuesday, Friday: 1:00 pm – 2:25 pm, Room: GITC 2315C

Instructor: Dr. Narain Gehani
Office: GITC 4304
Email: gehani@njit.edu
Office Hours: Friday: 11:00 am – 12:00 pm

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

Prerequisite: CIS 114 Introduction to Computer Science II (Data Structures)


Communication: Moodle (moodle.njit.edu) will be used for course communication. Please keep checking Moodle.

Assignments/Projects: Must be submitted on the due date in hard copy at the beginning of class.

Academic Integrity and NJIT Honor Code: Please familiarize yourself with the NJIT Honor Code (http://integrity.njit.edu). Violations of the honor code will be dealt with seriously and reported immediately to the Dean of Students.

Course Description

This course aims to teach database concepts and fundamentals as traditionally given to people learning how to build database systems. The focus is on understanding concepts as opposed to implementing algorithms that implement the concepts. For example, I will talk about B-tree indexes and discuss how they work and show you when and how to use them, but I will not discuss or show algorithms that implement B-trees.
The course will focus on relational databases covering the complete range of database topics. It will start from the basics such as discussing the need for databases, relational algebra, the relational database query language SQL, transactions, database integrity, triggers, indexes, design of good relations, views, database security, replication, logs, database design and tuning, and applications.

Projects
The course will require several practical projects to be done in teams. Included will be the writing of the project requirements.

Please form teams of 3 students to work on the team projects.

Database Used for the Course
We will primarily use the MySQL database for the course. However, we will also discuss the Oracle database later in the course.

Course Goals

Upon completion of the course, students will

- Understand relational database systems
- Learn SQL and be able to use it to manipulate a database
- Be able to design database tables and write database applications (the latter based on informal customer requirements), work in teams, and do team presentations.

Learning Outcomes

Students will learn the following (list numbering refers to ABET categories):

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline: the students apply knowledge of computing and mathematics in designing, implementing, testing, and assessing the quality of database systems.

c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs: The students work on a term project which involves analyzing requirements collected from prospective users of a system in order to design, implement, test, and evaluate a database system that satisfies the requirements.

d) An ability to function effectively on teams to accomplish a common goal: the students work in teams on a term-long project in order to design and implement a database system using a DBMS.

e) An ability to communicate effectively with a range of audiences: while working on the term-long project the students have to communicate and interact with the
members of their team. At the end of the semester, they have to demonstrate and make the case for the database system they have developed.

g) An ability to use current techniques, skills, and tools necessary for computing practice: The students apply current design techniques and a DBMS to implement an operational database system.

h) Theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices [CS]: In designing and implementing a database system the students learn the trade-off between space consumption, query performance, and maintenance cost.

i) An ability to apply design and development principles in the construction of software systems of varying complexity: The students apply the database design methodology presented in class to design a database system.

What do the Assignments Accomplish?

By doing the projects, students

1. will develop problem solving expertise,
2. be able to write database applications based on requirements that are not well-defined as happens in the real world,
3. learn and enhance knowledge their knowledge of SQL, and
4. develop team working skills (by doing team projects).

Specifically, having determined the application to be written, students will use SQL facilities such as table definitions, the SELECT-FROM-WHERE statement (to perform projections, selections, and joins), transactions, triggers, indexes, and build database application that address issues such as database security, replication, logs, database design and tuning. Students will also use the Java API (or some API of their choice) to access MySQL and provide a good user interface.

Course Grade Components

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.

Policy on Collaboration

Students found cheating, plagiarizing, or collaboration (collaboration is allowed for those working together in approved team projects) 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.

**Lectures**
(topic sequence is a guideline and may vary)

<table>
<thead>
<tr>
<th>Topics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction, Syllabus, Book; assignments</td>
<td></td>
</tr>
<tr>
<td>2 Project 0: get MySQL access / Database Models</td>
<td></td>
</tr>
<tr>
<td>3 Interacting with a database / Disk vs. Main Memory Databases</td>
<td></td>
</tr>
<tr>
<td>4 The Everest Books Database Requirements</td>
<td></td>
</tr>
<tr>
<td>5 MySQL: Single User vs. Client-Server</td>
<td></td>
</tr>
<tr>
<td>6 Relational Databases</td>
<td></td>
</tr>
<tr>
<td>7 The Everest Books Database Tables</td>
<td></td>
</tr>
<tr>
<td>8 SQL basics</td>
<td></td>
</tr>
<tr>
<td>9 Normal Forms</td>
<td></td>
</tr>
<tr>
<td>10 Relational Algebra</td>
<td></td>
</tr>
<tr>
<td>11 Everest Books Database Queries</td>
<td></td>
</tr>
<tr>
<td>12 JDBC</td>
<td></td>
</tr>
<tr>
<td>13 Transactions</td>
<td></td>
</tr>
<tr>
<td>14 Transactions / Locks &amp; Deadlock</td>
<td></td>
</tr>
<tr>
<td>15 Transaction Isolation Levels</td>
<td></td>
</tr>
<tr>
<td>16 Transactions in MySQL / Project 2 due</td>
<td></td>
</tr>
<tr>
<td>17 Project 3 Discussion / Project 3 start</td>
<td></td>
</tr>
<tr>
<td>18 Constraints</td>
<td></td>
</tr>
<tr>
<td>19 Triggers</td>
<td></td>
</tr>
<tr>
<td>20 Objects</td>
<td></td>
</tr>
<tr>
<td>21 Indexes</td>
<td></td>
</tr>
<tr>
<td>22 Views, spatial databases</td>
<td></td>
</tr>
<tr>
<td>23 Security</td>
<td></td>
</tr>
<tr>
<td>24 Logs &amp; Recovery</td>
<td></td>
</tr>
<tr>
<td>25 Tuning / Is your DB Truly Relational</td>
<td></td>
</tr>
<tr>
<td>26 Oracle</td>
<td></td>
</tr>
</tbody>
</table>
Some Important Dates

Midterm Exam Date

March 11, 2016 (Friday)

Final Exam Date

To be determined by the Registrar

Project Submission + Presentation Dates

<table>
<thead>
<tr>
<th>Assignment / Project</th>
<th>Section 2 Due Dates</th>
<th>Section 4 Due Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>February 5, 2016</td>
<td>February 5, 2016</td>
</tr>
<tr>
<td>2</td>
<td>February 26, 2016</td>
<td>February 26, 2016</td>
</tr>
<tr>
<td>3</td>
<td>March 25, 2016</td>
<td>March 25, 2016</td>
</tr>
<tr>
<td>4</td>
<td>April 22, 2016</td>
<td>April 22, 2016</td>
</tr>
</tbody>
</table>
MySQL Database @ NJIT

To get a new MySQL account (along with a database password) go to

ist.njit.edu/support/db/mysql.php

or

http://mypassword.njit.edu/db

Besides noting the password given to you, note the server (sql.njit.edu or sql2.njit.edu) assigned to you.

Command-line interface

1. Connect to afs1.njit.edu or afs2.njit.edu (afsconnect1.njit.edu or afsconnect2.njit.edu from outside)

   a. Windows Computers: Use SSH
      • At NJIT, Programs → SSH
      • For home use, download SSH from

         ist.njit.edu/software/index.php

   b. Linux Computers: Open the Terminal (the location varies by interface - KDE, GNOME, etc. Most often it will be in Applications or System Tools) and connect directly to the SQL servers via the "mysql" command.

2. Sign in and then type

   mysql -u UCID -p -h sql.njit.edu UCID

   You will need your database password and you may need to replace sql with sql1 or sql2 depending upon the server assigned to you.

Web interface

Go to

http://web.njit.edu/mysql/PhpMyAdmin/

You will need to use your database password.

Note: Your database name will be your UCID – you will not be able to create a new database
Oracle Database @ NJIT

To get an Oracle password, go to

http://mypassword.njit.edu/db

Select the option in “Database to change” that says

Oracle “course” DB on prophet.njit.edu

Command-line Interface

login to afs1.njti.edu using secure shell and enter at the prompt:

module load oraclient-12c

to load the Oracle client.

To activate the client type

aqlplus UCID@course

Now you are ready to interact with Oracle using SQL or PL/SQL.

Visual interface

Aqua Data Studio is a database query and administration tool that allows developers to create, edit, and execute SQL scripts, as well as browse and visually modify database structures.
A course on algorithms and data-structures. Advanced topics in data structures and algorithms, involving sequences, sets, and graphs such as searching, sorting, order statistics, balanced search tree operations, hash tables, graph traversals, graph connectivity and path problems. Algebraic and numeric algorithms. Performance measures, analysis techniques, and complexity of such algorithms. Greedy algorithms and dynamic programming-based techniques. String matching algorithms. Introduction to NP-completeness.

1.1 Contact Information

INSTRUCTOR: Alex Gerbessiotis
OFFICE: GITC 4213, 4th floor
OFFICE HOURS: Wed 4:00-5:30pm and Thu 4:00-5:30pm
ASSISTANT: TBA on course web-page
E-MAIL: alexg+cs435@njit.edu
TEL: (973)-596-3244
Else, by appointment Mon/Thu

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

1.2 Course Administration

Prerequisites CS 241, CS 288.


Course Work: 3 exams (including the final); Programming assignments; an unannounced quiz may be given (maybe after week 8) of 40 points; if I don't give you a quiz you get the points for free. If the Exam1 day is a snow day, then Exam1 will be held a week later; if that day is also a snow day it will be scheduled for April 14 as Exam 2.5, a 110pt or 150pt exam.

Grades: 1000 points = MP(160) + Ex1(110) + Ex2(345) + Ex3(345) + 40?

MP A programming mini-project (MP) with 3 options each one worth 120 points. A student may submit one or two options but NOT more; both options must be submitted as one email. No more than 160 points of credit from the sum of the grades of the two options. If a student receives a total of 60 points or less, 0 will be recorded towards the final grade; this implies a threshold of 25% for clearing the mini-project hurdle. See Handout 2 for more details and how to submit it. All options are due BEFORE noon of April 21, 2016; emails received at noon time or later will have 40pts deducted per option, and so done at noon time the following day(s), if applicable (i.e. there are points to deduct).

PS Approximately eight problem sets PS1-8 will be periodically posted along with their solutions. Exams may draw from these problem sets.

Exams Dates in Course Calendar. Exam 1 (or 2.5) is closed-everything. The other exams (midterm and final) are open-textbook only. You may bring a hard-copy of the textbook but you are not allowed to borrow one during the exam. For the final, you may also bring in class a clean copy of Handout 5 on red-black trees in addition to the textbook. Exam1 is on Thu Feb 11, 45min, 110 points. Exam2 is on Thu Mar 3, 2hrs, 345 points. Exam3 is on Thu May 12, 2hrs, 345 points.

Exam Conflicts This is a high-numbered required course. In case of multiple exams on a same day, this exam has priority even if it is the last exam of the day.
2.1 Course Objectives and Outcomes

Objective 1
Learn how to describe the asymptotic performance of algorithms and data structure operations.

Objective 2
Learn how to derive and determine the asymptotic performance of algorithms and data structure operations.

Objective 3
Learn how fundamental algorithms and data-structures operate, and understand their characteristics. Be able to choose among a variety of similar ones based on problem/program specification and requirements.

Objective 4
Learn how to compose more complex algorithms and data structures using as building blocks the fundamental algorithms and data structures introduced in class.

Objective 5
Learn how to compose more complex algorithms using the algorithmic design techniques introduced in class.

Outcome 1
Be able to asymptotically compare functions using o, O, ω, Ω, Θ, and be able to solve recurrences using the master, the iteration/recursion tree, and the substitution method.

Outcome 2
Become familiar with a variety of sorting algorithms and their performance characteristics (e.g., running time, stability, space usage) and be able to choose the best one under a variety of requirements.

Outcome 3
Be able to understand fundamental algorithms and data structures and be able to trace their operations for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs.

Outcome 4
Be able to identify the performance characteristics of fundamental algorithms and data structures for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs; be able to select among multiple available solutions to meet desired needs.

Outcome 5
Be able to understand fundamental algorithm design techniques and understand how to use them to design, implement and evaluate a variety of algorithmic problems.

Outcome 6
Be able to use the fundamental algorithms introduced in class to design, implement and evaluate algorithms for more complex problems.

Outcome 7
Be able to use current algorithmic techniques and skills for computing practice.

2.2 Topics to be covered

T1: AL1(2)/AL2(1)/AL3(1): Introduction, Algorithm Design Techniques (Incremental, Divide-and-Conquer)
T2: AL1(2)/AL2(1): Sorting Algorithms (Insertion, Selection, BubbleSort, MergeSort) Asymptotic growth of functions
T3: AL1(2)/BS(1): Recurrences
T4: AL3(1), BS(1): Brief Review on elementary data structures (Stacks, Queues, Trees, Lists)
T5: AL2(2), AL7(1): HeapSort, Priority Queues, Huffman Coding, and QuickSort (Worst-case and Average-case analysis)
T9: AL2(2), AL3(1): Dynamic Programming and Chained Matrix Multiplication, Arithmetic problems
T10: AL3(2), AL7(1): Union Find Algorithms; Introduction to Graph Algorithms
T11: AL3(3), AL7(2): Depth First Search, Breadth First Search, Minimum Spanning Trees.
T12: AL3(5): Shortest path Algorithms (Dijkstra and Floyd-Warshall)
Section 2.2 of the previous page contains a tentative list of topics that is intended to be covered in class. The code Tt refers to a topic. A topic may spread over one or more lectures. The code ALt refers to the ACM Computing Curricula 2013 topic description code. In parentheses, we provide an approximate number of hours per topic. Hour coverage may change depending on circumstances (e.g., class pace, weather). Minimum time requirements of the topics covered are: AL1 (basic analysis): 4, AL2 (algorithmic strategies): 6, AL3 (fundamental data structures and algorithms): 12, AL4 (basic automata, computability and complexity) optional: 2, AL7 (advanced data structures, algorithms and analysis) elective, DS3 (proof techniques): 1, DS5 (graphs and trees): 1.

3.1 Tentative Course Calendar

<table>
<thead>
<tr>
<th>Week**</th>
<th>Thu</th>
<th>PS with Solutions</th>
<th>MP</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>1/21</td>
<td>PS1*</td>
<td>MP out</td>
<td></td>
</tr>
<tr>
<td>W2</td>
<td>1/28</td>
<td>PS2*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W3</td>
<td>2/4</td>
<td>PS3*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W4</td>
<td>2/11</td>
<td>Exam1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W5</td>
<td>2/18</td>
<td>PS4*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W6</td>
<td>2/25</td>
<td>PS5*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W7</td>
<td>3/3</td>
<td>Exam2</td>
<td>Spring Break</td>
<td></td>
</tr>
<tr>
<td>W8</td>
<td>3/10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-</td>
<td>3/17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W9</td>
<td>3/24</td>
<td>PS6*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W10</td>
<td>3/31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W11</td>
<td>4/7</td>
<td>PS7*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W12</td>
<td>4/14</td>
<td>PS8*</td>
<td>MP in before noon</td>
<td></td>
</tr>
<tr>
<td>W13</td>
<td>4/21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W14</td>
<td>4/28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-</td>
<td>5/5</td>
<td></td>
<td></td>
<td>5/5 is a Reading Day</td>
</tr>
<tr>
<td>W15</td>
<td>5/12</td>
<td>Exam3</td>
<td>Final Examination</td>
<td></td>
</tr>
</tbody>
</table>

* Problem Sets (PS) with solutions are not for credit. ** In this calendar, a week starts on a Tuesday

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.

3.2 Absenteeism

Makeup It's up to you to make up for lost time; no make up if we give the quiz, for any reason (including DOSS justified one). No MP extensions for any reason, medical or otherwise; you have 3 months to submit it. 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. The maximum accommodation will be the number of missing days to the exam date. If you contact DOSS on the 3rd day and you are accommodated for just one, you will not be given a make-up. Let the instructor also know as soon as possible and simultaneously with DOSS.
Programs
Submitted code must conform to the requirements of Handout 2 and the MP. Code must be ANSI compliant and neither hardware-specific nor OS-specific or library-specific. Programming problems are graded based on test instances decided by the grader on a test platform of the grader's choice (e.g. afsconnect1.njit.edu). Do not expect partial credit if your code fails to run on all test instances unless you accompany your code submission with a DETAILED BUG REPORT.

Extensions
None: you are given 3 months to submit the MP.

Grading
Written work will be graded for conciseness and correctness. Use formal arguments, be brief and to the point, and write clearly. Material covered in class and appearing in the relevant notes and chapters of the designated textbook can be used without proof. DO NOT USE pencils to write down your solutions; if you decide to use a pencil do not complain about grading AFTER AN/THE EXAM.

Grades
Check the marks in written work and report errors promptly. Resolve any issue before the first Reading Day, or for the case of Exam3 and MP, 7 calendar days from the day grades were posted/made available. If you believe a grade you received for the solution of a problem is not representative of your effort, talk to the grader first and then to the instructor (if different). For programming assignments an email with your grade is sent back to you by replying to the email submission of the MP. The final grade is decided based on a 0 to 1000 point scale. Any student getting 25% (86 points) or less in the final exam will get an F, irrespective of other performance metrics. A student who otherwise collects at least 500 points and completes the minimum programming requirements should expect a grade of C or better; 850 points or more are usually required for an A but this threshold varies. The instructor reserves the right to push a student’s grade up based on that student’s significant programming effort.

Incomplete
A grade of I(incomplete) is given in rare cases where work cannot be completed during the semester due to documented long-term illness or absence (e.g. unexpected national guard duty). A student needs to be in good standing (i.e. passing the course before the absence) and receives a provisional I if there is no time to makeup for the documented lost time; a letter (or email) with a timeline of what is needed to be done will be sent to the student. Note that for most cases an I would be resolved withing few days, not months and not the following semester! Not showing up in the final will probably get you an F rather than an I.

Collaboration
Collaboration of any kind is NOT allowed in the in-class exams nor in the mini-project. Students who turn in code obtained through the Internet or otherwise, or is product of another person’s/student’s work (same or other section, same or other year, etc), risk severe punishment, as outlined by the University. The student will receive 0 in the MP and so will any collaborators (even unwitting ones). In addition to University penalties, the course grade will be reduced by one level (a C+ would become a C) for all parties involved. The work you submit must be the result of your own effort and you must safeguard it.

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

Email/SPAM
Always use NJIT email address; NJIT spam filters might be unpredictable. Include cs435 in the subject line. Do not complain otherwise.

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 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
Senior Project Capstone Courses  
CS/IT491  
(Spring 2016)

Course Identification:

Course:  Senior Project Capstone Course  
Instructor:  Osama Eljabiri, PhD  
Office:  Room 4210 - GITC Building – 4th Floor  
Office Hours:  Starting Feb [Thursdays 4:00-5:30PM (Spring 2016)/ Saturdays from 2-4PM by appointment only]  
Class-based extended office hours:  15-30 Minutes before most classes and up to one hour after most evening classes (as necessary).  
Live office hours: Online assistance and orientation available via email  
Telephone:  (973) 642-7123 or Cell Phone: (732) 456-0249 (preferred)  
FAX:  (866) 605-9416  
E-mail:  Osama.eljabiri@njit.edu or eljabiri@gmail.com  
Class Time/Location: As posted on the registrar system and MOODLE  
Virtual Classroom System:  http://MOODLE.njit.edu  (or moodle.njit.edu )

1. 1. Course Information:

A. Course Number, Title, Credits  
CS/IT 491, Senior Capstone Project, 3 credits.

B. Prerequisites  
Senior standing. An opportunity for students to integrate the knowledge and skills gained in previous information technology work into a real world team-based project. The project involves investigation of current literature as well as implementation of either a part of a large application or the whole of a small system.

C. Course Description  
The CS/IT Capstone Project is intended to provide a real-world project-based learning experience for seniors in the computer science undergraduate degree. The overall objectives of this course are to investigate the nature and techniques of a business and computing development project. Projects are either provided by industry partners or proposed by students who wish to become entrepreneurs. Entrepreneurship projects (E-teams) are intended to build a foundation for real world businesses. E-teams’ project proposals are reviewed by a panel of expert judges prior to approval. E-teams are mentored and evaluated by an entrepreneurship board of industry, university and alumni advisors. E-teams will carry out market research, target real world stakeholders and validate solutions using quantitative analysis based on customers’ feedback via questionnaires.

The course involves business analysis, business modeling, project management, feasibility analysis, risk analysis, R&D, requirements engineering, system design, implementation, quality assurance, documentation and presentation of a real world business problem and solution. The course is interdisciplinary in nature where students use their collective knowledge in business and technology to provide creative solutions in collaboration with real world project stakeholders.
2. Course Features and Objectives:

A- Features:
This course has unique features that are not currently offered through any other course on campus. These features are:

- It provides hands-on multidisciplinary real world experiences that integrate business applications with information technology areas such as multimedia and network security.
- It strengthens the 4-year college curriculum by enabling students to use what they learn collectively and creatively.
- It simulates the real-world environment internally in the structure of students’ teams and course “virtual organization”.
- It offers dynamic market-driven training that reflects hot topics highly demanded by industry but not usually covered through a static college curriculum.
- It enables students to master career-oriented skills such as leadership, presentation, entrepreneurship, social and communication skills.
- It shows how both IT and business knowledge are used to solve real-world IT problems.
- The experience gained working on such projects will make students more employable by industry including the ability of building businesses through the entrepreneurship track.

B- Specific goals for the course

Students who complete this course successfully will have:

- Ability to breakdown complex problems into manageable pieces (using WBS and Gantt).
- Ability to identify project risks and suggest strategies to minimize them.
- Ability to define project stakeholders, scope & requirements (including the use of FDD).
- Ability to capture, map and visualize the design of the proposed solution identifying key components and their relationships (examples: class diagram, ERM, network diagram, system architecture, etc.)
- Ability to implement the solution successfully using software and/or hardware technologies or other project-related tools (via prototypes).
- Ability to test (validate and verify) the quality of the executed solution using user feedback and other testing techniques.
- Ability to communicate a value proposition of the project to various stakeholders including the ability to explain, convince, engage and impress.
- Ability to organize the presentation in a meaningful and professional fashion including mastering personal and collaboration presentation skills.

Accordingly, the general outcomes of this course include:

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
(d) An ability to function effectively on teams to accomplish a common goal
(e) An understanding of professional, ethical, legal, security and social issues and responsibilities
(f) An ability to communicate effectively with a range of audiences
(h) Recognition of the need for and an ability to engage in continuing professional development
(i) An ability to use current techniques, skills, and tools necessary for computing practice.
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.
3. Course Details:

A. Course outline with approximate week-by-week schedule (subject to change if necessary)

**CS/IT Capstone Project Course**

*(Unless it is a combined event for both sections, the first date is always for the Tuesday section and the second date is for the Thursday section – Honors class will join regular teams)*

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>What to do</th>
<th>What to submit</th>
<th>What to Present?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>Course Introduction</td>
<td>Answer the survey</td>
<td>Capstone Survey Response</td>
<td></td>
</tr>
<tr>
<td>Jan 19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan 21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>Training Part 1 (manage and define): define requirements, manage project and apply SCRUNG + Capstone Open House Orientation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan 26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan 28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>Capstone Open House @ Campus Center Ballroom (Wednesday Feb 3th, 2015 2:30-6PM) Projects Demos, students apply, applications approved and teams formed</td>
<td>- Select a project or team</td>
<td>- Apply for project</td>
<td></td>
</tr>
<tr>
<td>Feb 3rd</td>
<td></td>
<td></td>
<td>- Apply for a job in a team</td>
<td></td>
</tr>
<tr>
<td>Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 4</td>
<td>Training Part 2 (design, develop, evaluate and present solution effectively): Navigate options, model solution, develop a prototype and test it, make presentation, receive feedback from judges – results announced in-class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 5</td>
<td>First Sprint Presentation / to sponsor only (Scope document and Gantt Chart) Project work in-team No regular class</td>
<td>Submit deliverables / present them</td>
<td>Post signed Scope and Gantt Chart on MOODLE as instructed</td>
<td>First Presentation to sponsors only</td>
</tr>
<tr>
<td>Feb 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb 18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 6</td>
<td>Project work in-team No regular class</td>
<td>Participate, contribute, and revise</td>
<td>Progress report (1) due on MOODLE by PM</td>
<td>None</td>
</tr>
<tr>
<td>Feb 24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feb 26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 7</td>
<td>Second Sprint Presentation / to sponsor only No regular class</td>
<td>Submit Prototype 1</td>
<td></td>
<td>Second Presentation to sponsors only</td>
</tr>
<tr>
<td>March 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>March 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 8</td>
<td>Project work in-team No regular class</td>
<td>Participate, contribute, and revise</td>
<td>- Progress report (2) due on MOODLE by PM</td>
<td>None</td>
</tr>
<tr>
<td>March 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>March 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week</td>
<td>Details</td>
<td>NO CLASSES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Week 9</strong>&lt;br&gt;March 22&lt;br&gt;March 24</td>
<td>Project work in-team&lt;br&gt;No regular class</td>
<td>Participate, contribute, and revise</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td><strong>Week 10</strong>&lt;br&gt;March 29&lt;br&gt;March 31</td>
<td>Project work in-team&lt;br&gt;No regular class</td>
<td>Participate, contribute, and revise</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td><strong>Week 11</strong>&lt;br&gt;April 5&lt;br&gt;April 7</td>
<td>MIDTERM PRESENTATIONS&lt;br&gt;Third Sprint Presentation / to class and sponsor&lt;br&gt;At the Faculty Dining Room – CTR Building 3rd floor</td>
<td>Submit deliverables and present them With Prototype 2&lt;br&gt;Invite sponsor&lt;br&gt;Sponsor evaluation is 30% of your midterm grade &lt;br&gt;Sprint (4) starts</td>
<td>Third Presentation to class MUST BE POSTED ON MOODLE THE NIGHT BEFORE</td>
<td></td>
</tr>
<tr>
<td><strong>Week 12</strong>&lt;br&gt;April 12&lt;br&gt;April 14</td>
<td>Fourth Sprint Presentation / to sponsor only&lt;br&gt;No regular class</td>
<td>Participate, contribute and revise&lt;br&gt;Sprint (5) starts</td>
<td>Fourth Presentation to sponsors only</td>
<td></td>
</tr>
<tr>
<td><strong>Week 13</strong>&lt;br&gt;April 19&lt;br&gt;April 21</td>
<td>Project work in-team&lt;br&gt;No regular class</td>
<td>Showcase poster and final presentation PPT are due</td>
<td>Showcase poster and final presentation PPT are due</td>
<td></td>
</tr>
<tr>
<td><strong>Week 14</strong>&lt;br&gt;April 26&lt;br&gt;April 28</td>
<td>Project work in-team&lt;br&gt;No regular class</td>
<td>Showcase poster and final presentation PPT are due</td>
<td>Showcase poster and final presentation PPT are due</td>
<td></td>
</tr>
<tr>
<td><strong>Week 15</strong>&lt;br&gt;Tuesday&lt;br&gt;May 3rd, 2016&lt;br&gt;4PM-9PM (CS, IS &amp; IT)</td>
<td>Fifth Sprint Presentation in Capstone Project Public Showcase @ Campus Center Atrium&lt;br&gt;May 3rd, 2016 4:00-9:00PM</td>
<td>Submit deliverables and present them With Final Product Release&lt;br&gt;-Full documentation Final Product All on one MOODLE</td>
<td>Final (5th) Project Presentation (5 minutes) @ showcase Table</td>
<td></td>
</tr>
<tr>
<td><strong>Week 16</strong>&lt;br&gt;The week after the showcase</td>
<td>Project work in-team&lt;br&gt;No regular class</td>
<td>Participate, contribute and revise&lt;br&gt;Final report due</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

Provide sponsor with link and team information Sponsor online evaluation due May 10th<br>Submit final report on MOODLE – if NOT waived
B. References

C. Number of hours of lecture, recitation, and laboratory

Students work on real-world entrepreneurial, industry sponsored or research development projects for the entire 14 weeks of the semester. The problem solving process is broken down into five “Sprints”. A sprint is a 2-3 week time-boxed duration or project phase that allows students and project ‘s stakeholders to manage change, risk and complexity and adapt to skillsets and project requirements through evolutionary prototyping. Lectures and training will include a comprehensive crash course in the few weeks and some on-demand training (when available) throughout the semester. Starting from the third or fourth week of the semester, students will be required to have frequent team meetings (at least once a week face to face) and frequent client meetings (at least once every two weeks). Additional hands-on training, project management training and laboratory hours will also be included.

GRADING POLICIES

Your final grade in this course will be based on the percentage of points that you receive out of the total possible points for the course (1100). Grades will be determined according to the following scale:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% - 100%</td>
<td>A</td>
</tr>
<tr>
<td>85% - 89%</td>
<td>B+</td>
</tr>
<tr>
<td>80% - 84%</td>
<td>B</td>
</tr>
<tr>
<td>75% - 79%</td>
<td>C+</td>
</tr>
<tr>
<td>70% - 74%</td>
<td>C</td>
</tr>
<tr>
<td>60% - 69%</td>
<td>D</td>
</tr>
<tr>
<td>0% - 59%</td>
<td>F</td>
</tr>
</tbody>
</table>

Capstone Evaluation Criteria
Spring 2016

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Percentage</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attendance</td>
<td>By deduction</td>
<td>3% deduction for every class absence</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2% deduction for each other group meeting absence</td>
</tr>
<tr>
<td>Capstone Training</td>
<td>5% (50 points)</td>
<td>Attending training sessions</td>
</tr>
<tr>
<td>Item</td>
<td>Percentage</td>
<td>Details</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Sprint 1 Deliverables online</td>
<td>4% (40 pts)</td>
<td>Sprint 1 Deliverables: Signed scope document and Gantt Chart uploaded to MOODLE</td>
</tr>
<tr>
<td>Midterm Presentation</td>
<td>15% (150 pts)</td>
<td>In-class Midterm Presentation 5% of midterm grade is based on sponsor feedback</td>
</tr>
<tr>
<td>2 Sprints presentations to sponsors</td>
<td>6% (60 pts)</td>
<td>2 x 50 = 100 points (Sprint 2 and 4) - Verified by progress report submitted by team leader on MOODLE</td>
</tr>
<tr>
<td>Full Attendance and Participation at the final Showcase</td>
<td>5% (100 pts)</td>
<td>Full attendance from set-ups to clean-ups Excellent table/poster preparation</td>
</tr>
<tr>
<td>Final product (Evaluated by Sponsor)</td>
<td>30% (300 pts)</td>
<td>Final working IT or software solution (Tested and installed, with full source code) (Hard and software copy required)</td>
</tr>
<tr>
<td>Final presentation (evaluated by judges at Showcase)</td>
<td>20% (200 pts)</td>
<td>Present both MDDDE and solution implementation</td>
</tr>
<tr>
<td>Final Project CD (Evaluated by instructor)</td>
<td>5% (50 pts)</td>
<td>Your CD should include: - Refined/expanded/updated final PPT of the project according to MDDDE - Short and Long CAMTESIA - An electronic copy of your poster</td>
</tr>
<tr>
<td>Final Report OR Second Project OR Capstone CISCO Academy (when available)</td>
<td>10% (100 pts)</td>
<td>Your choice of one of three options that must be completed successfully and verified by deliverable or feedback</td>
</tr>
</tbody>
</table>
| Extra Credits                                                        | Up to (100 points) 10% | Leading a team successfully  
|                                                                      |            | Being an effective part of the executive team  
|                                                                      |            | Second project can waive final report requirement. Your final grade will be split 50/50 between the two projects but both will be full projects time. |
| Total                                                                | Max Total  | 1000 points 1100 points |

Please note that:
- Class attendance, and in-Class/online participation and collaboration is very important
- In-group participation and attendance is extremely significant in determining your final letter grade.

Good Luck,
Dr. Osama Eljabiri
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 programming methodology and problem solving. Topics include program design techniques, introductory data structures, and algorithms and their analysis. The Java programming language is fully discussed and serves as the vehicle to illustrate many of the concepts.

TEXTBOOK AND MATERIALS


COURSE OUTCOMES

a. Gain familiarity and comfort with the basic functionalities of Java.
b. Represent data using the appropriate data type.
c. Evaluate a situation in order to choose the appropriate selection structure.
d. Evaluate a situation in order to choose the appropriate iterative structure.
e. Use pre-existing and programmer defined classes.
f. Process homogeneous data.
g. Determine when to use inheritance to facilitate software re-use.
h. Determine when to use exception handling to handle erroneous situations.
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

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>January 19 – January 25</td>
</tr>
<tr>
<td>2</td>
<td>January 26 – February 1</td>
</tr>
<tr>
<td>3</td>
<td>February 2 – February 8</td>
</tr>
<tr>
<td>4</td>
<td>February 9 – February 15</td>
</tr>
<tr>
<td>5</td>
<td>February 16 – February 22</td>
</tr>
<tr>
<td>6</td>
<td>February 23 – February 29</td>
</tr>
<tr>
<td>7</td>
<td>March 1 – March 7</td>
</tr>
<tr>
<td>8</td>
<td>March 8 – March 21</td>
</tr>
<tr>
<td>9</td>
<td>March 22 – March 28</td>
</tr>
<tr>
<td>10</td>
<td>March 29 – April 4</td>
</tr>
<tr>
<td>11</td>
<td>April 5 – April 11</td>
</tr>
<tr>
<td>12</td>
<td>April 12 – April 18</td>
</tr>
<tr>
<td>13</td>
<td>April 19 – April 25</td>
</tr>
<tr>
<td>14</td>
<td>April 26 – May 5</td>
</tr>
</tbody>
</table>

LATE WORK AND MAKE-UP EXAMS

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.

SHARING INFORMATION

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.

**STUDENT CONDUCT**

The NJIT University code on academic integrity will be followed in all courses.

**STUDENT WITH DISABILITIES CODES**

NJIT adheres to section 504 of the Rehabilitation Act (ADA) of 1990. 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: 1-888-789-0407 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 Change mechanism. You will need to know your current UCID and UCID password. Questions can be referred to 1-973-596-2900.

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

**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
- Email
- 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.rjitu.edu
877-515-6096, press 2

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

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: Tuesday, December 1, 2015, 3:42 PM

Moodle Docs for this page

2016 Spring - CS 505552
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: Java Programming
Syllabus, Spring 2016

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

Fall 2016 Office Hours: Tuesday 1:00-2:30, Friday 2:30-4:00, or by appointment

CS 602 – Java Programming
Advanced Web-based programming with an emphasis on the Java language and platform. Basic constructs and syntax and then the core advanced features. Topics include: networking and sockets, remote method invocation (RMI), database connectivity (JDBC), Java Beans, multi-threading and lightweight components (Swing). Common gateway interface (CGI) languages and browser scripting (JavaScript and VBScript) are discussed when used as a complement to the functionality of the Java language.

Please include CS602 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.

Required Background: http://web.njit.edu/~gwryan/CS602

Course Webpage: http://web.njit.edu/~gwryan/CS602
All course materials, including lecture notes, assignments and solutions, will be posted on the web page. Announcements and notices will also be sent by email.

Textbook:

Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programs</td>
<td>40%</td>
</tr>
<tr>
<td>Midterm</td>
<td>27%</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:
- Java Overview and comparison to C++
- Basic Language Features: Primitives, Objects, Constructors, Variables, Methods, Classes, Access Specification, Inheritance
- Essential Java Classes
- Operators; Sequence, Selection and Repetition; Exception Handling; Inner Classes
- Interfaces; Event Handling; Layout Managers
- Abstract Window Toolkit (AWT) Event Handling
- Streams
- Swing
- Multithreading
- Networking
- Remote Method Invocation (RMI)
- Java Database Connectivity (JDBC)
- Java Beans; Applets v. Applications
- Security

Ethical Conduct

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 without attribution. You may NOT use code in your assignments that you did not write without attribution. Giving your work to someone else to copy from is just as much cheating as copying someone else’s work.

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 assignments will be in Java. There may be a project that involves HTML and/or Javascript as well.

- You will NOT need to submit a printout of your code.

- 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

- Every file should include a block of identifying comments at the very top of each file, as follows:
  - CS 602, the section number, and SPRING 2015
  - Your name
  - Assignment #

- 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.

<table>
<thead>
<tr>
<th>Important Dates</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9-Feb</td>
<td>Program 1 Due</td>
</tr>
<tr>
<td>1-Mar</td>
<td>Program 2 Due</td>
</tr>
<tr>
<td>1-Mar</td>
<td>Midterm</td>
</tr>
<tr>
<td>29-Mar</td>
<td>Program 3 Due</td>
</tr>
<tr>
<td>26-Apr</td>
<td>Program 4 Due</td>
</tr>
<tr>
<td>6-May - 12-May</td>
<td>FINALS WEEK</td>
</tr>
</tbody>
</table>

Note that exams are cumulative.
CS 610 Syllabus  
Spring 2016, Sect 106  

Data Structures & Algorithms  
T 6-9:05, TIER Lect 2  

Dr. David Nassimi

Prof.: David Nassimi  
Web: http://www.cs.njit.edu/~nassimi/cs610  
Email: nassimi@cs.njit.edu  
Tel: 973-596-5645; Office: GITC 4308  
Hours: T, R 4:00-5:30 pm  

TA: TBA  
Website:  
Email:  
Office:  
TA Hours:  

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

Prerequisites:  
1. Undergrad course on Data Structures & Algorithms (CS 505 or equivalent);  
2. Discrete Math (CS 506 or CS 241 or equivalent);  


Course Objectives (what you are expected to get out of this course):  
1. Learn basic analysis techniques  
2. Learn basic design techniques  
3. Review of induction and recursion, and proof techniques  
4. Learn recurrence equations and how they are used in analysis of algorithms  
5. Learn advanced data structures: Priority queues, heaps, hash tables, and search trees  
6. Understand sorting algorithms and their complexities  
7. Learn basic graph algorithms and their applications

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Exam Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments (5)</td>
<td>30%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>35% Week 8</td>
</tr>
<tr>
<td>Final</td>
<td>35% Finals week</td>
</tr>
<tr>
<td></td>
<td>Tues March 8</td>
</tr>
<tr>
<td></td>
<td>Tues May 10</td>
</tr>
</tbody>
</table>

Notes: Pictured NJIT ID required for all exams. All exams are closed books and closed notes. Average score of all exams must be at least 50% to pass the course.

Website: Class handouts (syllabus, assignments), announcements, and previous exams are posted. You are responsible for checking the website regularly for assignments/announcements.

Academic Integrity: Familiarize yourself with NJIT Honor Code: http://integrity.njit.edu. Any evidence of dishonesty will be dealt with seriously and reported to the Dean of Students.
Policies:
1. Assignments must be done by you individually (no team-work).
2. Assignments must be handed in (HARD COPY) at the start of the class period on the due date. Late assignments will not be accepted.
3. Programming assignments must be in C, C++, or JAVA.

### CS 610 Course Outline

<table>
<thead>
<tr>
<th>Week (Approx)</th>
<th>Topic</th>
<th>Reading (Goodrich)</th>
</tr>
</thead>
</table>
| 1             | Introduction, Analysis Techniques  
Examples of worst-case and average-case analysis  
Complexity definitions: O(), Omega, Theta | Ch. 1                            |
| 2             | Induction, Recursive Algorithms, Recurrence Relations,  
Divide-and-Conquer Technique and Recurrences  
Examples: Binary Search, Mergesort | p. 12; Sec. 4.1; Sec. 5.2  
Posted Notes |
| 3             | Self-Review: Lists, Stacks, Queues, Trees                           | Ch. 2, Sec. 2.1-2.3              |
| 4             | Advanced Data Structures  
Priority Queues, Heaps | Ch. 2, Sec. 2.4-2.5              |
| 5             | Sorting and Selection Algorithms  
Lower-Bound on Sorting by Comparison  
Insertion-Sort, Bubble-Sort, Selection-Sort  
Mergesort, Heapsort, Quicksort | Ch. 4                            |
| 6             | Integer Sorting: Bucket-Sort, Radix Sort  
Selection (Kth smallest element)  
Union-Find Algorithm (if time permits) | Sect. 4.2 (pp. 225-234)          |
| 7             | Dictionary ADT (Search, Insert, Delete)  
1 Binary Search Trees (BST); Average Analysis of BST  
2 Balanced Search Trees (AVL)  
3 Hash Tables | Ch. 3                            |
| 9             | Algorithms Design Techniques:  
Divide-and-Conquer  
Greedy (Examples: TSP, Huffman Coding)  
Dynamic Programming (Example: Robot Walk) | Ch. 5  
Sect. 9.3 (pp. 440-442) |
| 10-13         | Graph Algorithms  
Definitions, Representations  
Traversals  
Connected Components  
Single-Source-Shortest-Paths (Dijkstra)  
All-Pairs-Shortest-Paths (Floyd)  
MST Algorithms: Prim, Kruskal, Baruvka | Ch. 6, 7 |
| 14            | Text Processing (if time permits) | Ch. 9                            |
Intensive study of the fundamentals of data structures and algorithms. Presents the definitions, representations, processing algorithms for data structures, general design and analysis techniques for algorithms. Covers a broad variety of data structures, algorithms and their applications including linked lists, various tree organizations, hash tables, strings, storage allocation, algorithms for searching and sorting, and a selected collection of other algorithms.

1.1 Contact Information

**INSTRUCTOR:** Alex Gerbessiotis  
**E-MAIL:** alexg+cs610@njit.edu  
**OFFICE:** GITC 4213, 4th floor  
**TEL:** (973)-596-3244  
**OFFICE HOURS:** Wed 4:00-5:30pm and Thu 4:00-5:30  
**Else, by appointment Mon/Thu**  
**ASSISTANT:** TBA on course web-page  
**CLASS HOURS:** Wed 6:00-9:05pm, Culm LEC1  
**COURSE WEB PAGE:** http://www.cs.njit.edu/~alexg/courses/cs610/index.html

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

1.2 Course Administration

**Prerequisites**  
CS 505 or CS 335, and completion of all bridge course requirements. (The 335 is a relic from the past; the 505 is a not very active course either.)

**Textbook**  

**CourseWork:** 3 exams (including the final). Programming project. Potentially one more unannounced quiz (To be determined)?

**Grading scheme:** 1000 points = PP(134) + Ex1 (200) + Ex2 (333) + Ex3(333).

**PP**  
A programming project (PP) with 2 options each one worth 134 points. A student may submit one or two options but both options must be submitted in one shot; a grade of 0-40 points for an option is accounted as 0 towards the final grade. No more than 134 points of credit from the sum of the grades of the two options. See Handout 2 for more details.

**Practice PS**  
Four comprehensive problem sets PS1-4 will be periodically posted along with their solutions. Exams may be based on these problem sets.

**Exams**  
Dates in Course Calendar; all exams in classroom. Exams are open-textbook only; you may bring a hardcopy of the textbook but you may not borrow one during the exam. Exam1 (quiz) is on Feb 10, 60mins, 200 points. Exam2 (midterm) is on Mar 2, 2hours 333 points. Exam3 (final) is on May 11, 2hours, 333 points. A second quiz might be given around Apr 13; in that case Exam1’s grade gets halved and this quiz will count for the cut 100 points. Or if there is a snowstorm and class cancellation on Feb 10, Exam1 may get rescheduled on Apr 13.

**Due Dates**  
PP submitted only by email and MUST be received BEFORE NOON-TIME; 40 points subtracted from every option (after work is graded) at deadline plus 1 minute, and every 24-hour period thereafter. We strongly recommend that you use an NJIT email account to email your code; if you do not use one, do not complain about potential loss of emails (or NJIT’s blocking of them). Read Handout 2 for details and the homework itself. Programs can be done in C, C++, or Java and will be compiled and tested on afsconnect1.njit.edu or afsconnect2.njit.edu.
2.1 Course Objectives and Outcomes

Objective 1 Understand and formulate the input-output relationship of computational problems, and formulate the requirements, data and operations of abstract data types (ADT). data structures.

Objective 2 Learn, Understand and be able to describe data structures that represent the mathematical model underlying an ADT.

Objective 3 Learn how to describe, derive and determine, the asymptotic performance of algorithms and data structures.

Objective 4 Learn how fundamental algorithms and data-structures operate, and understand their characteristics. Be able to choose among a variety of similar ones based on problem/program specification and requirements.

Objective 5 Learn how to compose more complex algorithms using as building blocks the fundamental algorithms introduced in class.

Objective 6 Learn how to compose more complex algorithms using the algorithmic design techniques introduced in class.

Outcome 1 Be able to accurately specify the input/output relationship of computational problems, and describe correctly fundamental ADTs.

Outcome 2 Be able to accurately specify the operations of fundamental ADTs and the data structures underlying them; be able to correctly describe the performance of algorithms for the operations on these data structures.

Outcome 3 Be able to asymptotically compare functions using $o, O, \omega, \Omega, \Theta$.

Outcome 4 Be able to solve recurrences using the master, the iteration/recursion tree, and the substitution methods.

Outcome 5 Become familiar with a variety of algorithms for sorting, selecting and searching data and their performance characteristics (eg, running time, stability, space usage) and be able to choose the best one under a variety of requirements.

Outcome 6 Be able to understand fundamental algorithms and data structures and their performance and be able to trace their operations for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs.

Outcome 7 Be able to understand fundamental algorithm design techniques and understand how to use them to solve algorithmic problems.

Outcome 8 Be able to use the fundamental algorithms introduced in class to design algorithms for more complex problems and analyze their performance.

Outcome 9 Be able to use the design techniques and algorithms introduced in class to design algorithms for more complex problems and analyze their performance.
### 3.1 Tentative Course Calendar

<table>
<thead>
<tr>
<th>Week</th>
<th>Wed</th>
<th>Exams</th>
<th>PP or PS</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>W01</td>
<td>1/20</td>
<td></td>
<td>POut, PS1*</td>
<td></td>
</tr>
<tr>
<td>W02</td>
<td>1/27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W03</td>
<td>2/03</td>
<td></td>
<td>PS2*</td>
<td></td>
</tr>
<tr>
<td>W04</td>
<td>2/10</td>
<td>Exam1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W05</td>
<td>2/17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W06</td>
<td>2/24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W07</td>
<td>3/02</td>
<td>Exam2</td>
<td></td>
<td>aka Midterm</td>
</tr>
<tr>
<td>W08</td>
<td>3/09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-</td>
<td>3/16</td>
<td></td>
<td></td>
<td>Spring Break</td>
</tr>
<tr>
<td>W09</td>
<td>3/23</td>
<td></td>
<td>PS3*</td>
<td></td>
</tr>
<tr>
<td>W10</td>
<td>3/30</td>
<td></td>
<td></td>
<td>Mar28 is Last Day to Withdraw</td>
</tr>
<tr>
<td>W11</td>
<td>4/06</td>
<td></td>
<td>PS4*</td>
<td></td>
</tr>
<tr>
<td>W12</td>
<td>4/13</td>
<td></td>
<td></td>
<td>Pop-up quiz maybe?</td>
</tr>
<tr>
<td>W13</td>
<td>4/20</td>
<td></td>
<td>PPin</td>
<td></td>
</tr>
<tr>
<td>W14</td>
<td>4/27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W-</td>
<td>5/04</td>
<td></td>
<td>No-class</td>
<td>Reading Day</td>
</tr>
<tr>
<td>W15</td>
<td>5/11</td>
<td>Exam3</td>
<td></td>
<td>Final Exam</td>
</tr>
</tbody>
</table>

* Problem Sets (PS) with solutions are not for credit. The following describes a tentative list of topics that is intended to be covered in class with indicative chapter pointers to GT. The lecture summaries contain detailed correspondence to chapters of the textbook.

### Topics to be covered

**T1 : Ch1.5.2**
- Introduction (insertion-sort, fibonacci sequences).
- Algorithm Analysis (Asymptotic Growth of functions, recurrences)

**T2 : Ch1.4.1,5.2**
- Algorithm Design Techniques (Incremental, Divide-and-Conquer)
- Sorting (selection-sort, bubble-sort, merge-sort).

**T3 : Ch2.1-2.3**
- Elementary Data Structures and Trees. Tree traversals.
- Ch4.2 Union-find operations.

**T4 : Ch2.4,9.3**

**T5 : Ch4**

**T5 : Ch2.5-2.7**
- Hashing.

**T6 : Ch3**
- Binary Search Trees and Balanced Binary Search trees.
  - m-way trees, 2-3-4 trees, B-trees.

**T7 : Ch5**
- Integer operations (addition and multiplication).
  - Matrix operations (addition and multiplication). Strassen’s method.
  - Dynamic Programming and chained matrix multiplication.

**T8 : Ch6**
- Graphs and their representation. Graph traversals (DFS,BFS).
  - Strongly connected components. Topological sorting.

**T9 : Ch7**
- Weighted graph problems. Shortest-path problems (Dijkstra’s).
  - All-pairs shortest paths and transitive closure (Floyd-Warshall).
  - Spanning trees (Prim’s and Kruskal’s algorithms).

**T10 : Ch9**
- String and Pattern matching algorithms.

**T11 : Ch10**
- Fundamental algorithms involving numbers. RSA, FFT.

**T12 : Ch13**
- P and NP, NP-completeness.

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.
MISSING

If you miss a class, it's up to you to make up for lost time. If you miss Exam1 and the DEAN OF STUDENT SERVICES certifies your absence after receiving valid documentation from you, then the scaled Exam2 grade will be used. You MUST CONTACT DOSS within 2 working days from the day the reason for the absence is lifted. If you miss the unannounced pop-up quiz and you have a valid (DOSS granted) excuse for a make-up, Exam1 stays as is (and no make-up exam will be given).

Program

Check Handout 2. Each program option is evaluated/graded first. Then the lateness penalty, if any, is applied. If the resulting grade for an option is 0-40 then the grade for that option is accounted as 0. The sum of the grades for the two options cutoff at 134 points is your programming grade.

Grading

Written work will be graded for conciseness and correctness. Use formal arguments. Be brief and to the point and write clearly. Material covered in class and appearing in the relevant notes and chapters of the designated textbook can be used without proof or citation. Programming points discarded may be used, at the discretion of the instructor, to remove a student from a fail zone. This assumes no violation of the Collaboration policy below. DO NOT USE pencils to write down your solutions; IF YOU DO USE PENCILS do not complain later about grading.

Grades

Check the marks in a written work and report errors promptly. Make sure you report such problems to the grader or the instructor within two weeks from receipt/return and no later than the (first) Reading Day whichever comes first. First resolve issues with the grader, then the instructor (if different). The final grade is decided on a 0 to 1000 point scale. A student who collects at least 500 points (including some minimal programming work) should expect a passing grade (C or better). 900 points or more are usually needed for an A including robust programming work. (Assuming always no violation of the Collaboration policy.)

Incomplete

A grade of I(incomplete) is given in rare cases where work cannot be completed during the semester due to documented long-term illness or absence (e.g. unexpected national guard duty). A student needs to be in good standing (i.e. passing the course before the absence) and receives an I if there is no time to make up for the documented lost time. Not showing up in the final will probably get you an F rather than an I.

Collaboration

Collaboration of any kind is PROHIBITED in the in-class exams or the project. A student must turn in code that has fully been written by him/her. Any submitted code (even few lines) obtained through the Internet or otherwise, or is product of someone else's work or is common with another student submission, in the same or other section/course, risks severe punishment, as outlined by the University; all parties of such interaction receive automatically 0 and grade is lowered by one or two levels. The work you submit must be the result of your own mental effort and you must safeguard it from other parties; if you can't protect your home computer, use an AFS machine.

Mobile Devices

Mobile devices etc must be switched off/shut before and during class exams or lectures.

Email/SPAM

Use an NJIT email address or your email might not reach us. Send email to the designated course email address and always include cs610 in the subject line.

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 632: Advanced Databases

Faculty Contact Information
Instructor: James Geller
Email: james.geller@njit.edu
Office Hours: E-mail directly if you have any questions

Course Description
This course has four parts:
1. Review your Oracle SQL and learn more of it.
2. Learn Oracle 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 with Oracle).
4. Study a few advanced topics on databases, such as XML, OODBs, DB security, parallel and distributed databases, No-SQL databases, indexing 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.

Textbook and Materials
- RECOMMENDED: Page-Jones, M. Fundamentals of Object-Oriented Design in UML. Addison-Wesley, 1st edition, 1999, ISBN 978-0201699463. (I highly recommend that students buy books. Having books on Java, HTML, XML, and JavaScript is a good idea, even if you will not use them a lot or use the Web.)
- WEB MATERIAL: There will be materials posted on my Web page. Details will be provided during the course.

Course Outcomes
This course covers the following concepts and principles:
- PL/SQL as "Normal" Language
- PL/SQL Accessing Tables
- Using AQUA for Oracle Programming
- Object Oriented Modeling
- Inserting etc. in PL/SQL
- Object-Oriented DBs
- Triggers in PL/SQL
- Object-Relational DB
- Objects in SQL
- Introduction to UNIX/LINUX Use
- Intro to XML
- Database Security
- XML in Oracle SQL
- Querying XML
- Distributed Databases
- PSP Programming
- Web Databases
- No-SQL Databases
- Indexing and Performance Tuning

### Grading Categories

<table>
<thead>
<tr>
<th>Categories</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm Exam</td>
<td>100 Points</td>
</tr>
<tr>
<td>Final Exam</td>
<td>200 Points</td>
</tr>
<tr>
<td>Project 1</td>
<td>40 Points</td>
</tr>
<tr>
<td>Project 2</td>
<td>40 Points</td>
</tr>
<tr>
<td>Project 3</td>
<td>40 Points</td>
</tr>
<tr>
<td>Project 4</td>
<td>50 Points</td>
</tr>
<tr>
<td>Homework Questions (13 Lectures: 5 questions @ 2 points each)</td>
<td>130 points</td>
</tr>
<tr>
<td>TOTAL POINTS</td>
<td>600 points</td>
</tr>
</tbody>
</table>

At the end of the semester, I will add up your total points and curve the results for the whole class. There will be no letter grades and no curves during the semester.

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 or most of the points on the Project 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.

Exams:
There will be one Midterm Exam and one Final Exam. **Exams must be taken at an approved testing center with a proctor.** (Or at NJIT).

Projects:
There will be four project assignments consisting mostly of SQL and 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.

Homework:
There will be 5 graded and additional ungraded homework questions in every single lecture. Answers will be usually short, sometimes a single word, sometimes a sentence.

---

**Course Structure**

<table>
<thead>
<tr>
<th>Lecture 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 DB</td>
</tr>
<tr>
<td>Objects in SQL</td>
</tr>
<tr>
<td>Introduction to UNIX/LINUX Use</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>Web Databases</td>
</tr>
<tr>
<td>No-SQL Databases</td>
</tr>
</tbody>
</table>
Indexing and Performance Tuning

A brief explanation: Why are there Modules, Lectures, and Parts?

The course CS632 is offered both during the summer and during the semester. The summer term is more intensive and only 10 weeks are available. Thus, in summer each week corresponds to one lecture and each lecture has 5 parts. Each part is a half-hour long video recording.

Altogether there are 51 parts (10 weeks times 5 parts, plus a final review part).

During the semester we have 14 weeks time. Thus, during the semester you need to watch only 4 parts (4 half hours) per week. That gives you more time for digestion. After 12 weeks that comes to 48 parts, and in the 13th week there are 3 more half hour lectures, adding up to 51 again.

The 14th week is for you to implement the last project and prepare for the final exam. No new material is covered in the 14th week.

Another brief explanation: Why are there only two hours of lecture per week during the semester? Isn't the lecture three hours when you take it on campus?

The video recordings cover exactly the same amount of material as is covered on campus. On campus a lot of time gets spent on collecting, explaining and handing out projects, students walking in late and disrupting the class, students asking questions during the class, explanations of midterm exams (after the exam), handing out additional account information, etc.

Furthermore, face to face, the midterm material is broken into two midterms. As it is difficult for Distance Learning students to come to exams, your class will have only one midterm. This exam takes up one additional lecture (2.5 to 3 hours) in the class room.

Time Commitment

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

Computer Use in the Course

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).

Oracle and Aqua:
We are using only the Oracle database system and PL/SQL. PL/SQL is part of the normal Oracle distribution.

We are using Aqua Data Studio as interface to Oracle. Here is information on Aqua. You may download Aqua Data Studio from this URL: http://ist.njit.edu/software/download.php

Aqua Activation:
Please note that access to this commercial software is highly restricted. At the time of
this recording we are using Aqua 12. However, by the time you are taking the class, it might already be a newer version. However, as we don't use sophisticated features, the versions are almost equivalent for us. This is no reason for concern. You need to follow the setup steps exactly, or it won't work. Exactly, http://ist.njit.edu/software/ads/12/Activate.php

Originally, Aqua is from: http://www.aquafold.com/downloads.html

You may find documentation for Aqua Data Studio from this URL: https://www.aquachusters.com/app/home/project/public/aquadataspotmedical/wikibook/Documentation11/page/0/Aqua-Data-Studio-11-0
This is for version 11, and you are downloading a higher version right now.

NJIT Passwords:
If there is an NJIT database password problem, try this: 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/

Oracle:
The ORACLE login will ask you for your AFS ID, password, and connection identifier. You need a separate password for Oracle. You get it yourself at: https://mypassword.njit.edu/db
Even though you never had a password, the procedure for getting a new password is the same as if you already had a password and you are resetting it. Passwords are valid for one year.

The computer we are using is prophet.njit.edu
The connection identifier that you are asked for (much later) is course.

Furthermore, you can get extensive explanations of the ORACLE set up 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 much later.

Sharing Information
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 zero (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 (except the professor). 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) Website. 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 reduce your homework credit to zero, from whatever it was before.

Let’s Talk About Email

I want to start off by reminding you there are about 36 of you and only one of me. If every one of you sends me two emails per day, seven days a week and it takes me five minutes to read and intelligently respond to one email that will be 42 hours in total. By “the rules” I am required to work about nine hours per week for a class (and I do much more). However, 42 hours won’t work. During the work week, I am trying to respond to email (or at least acknowledge email) within 24 hours. If the number of messages explodes, I can’t promise it. Therefore, please adhere to the following rules to make my life easier (and save you frustration too).

1. Every message you send me has to have a subject line that starts with **CS632**.
2. Every message has to be signed **with first name and last name**. I don’t know who bs349@njit.edu is.
3. Try not to send me email from multiple email accounts. Use only the NJIT account.
4. Please don’t send me a message “did you get my email?” on the same day. If everybody does that, you are doubling the number of messages I get per day. Wait 24 hours.
5. While I am making an attempt to reply to email over the weekend, and I often do, I **do not promise it**.
6. When I am on a trip (conference, etc.), I will warn you in advance and I will put up a “vacation message.” In that case I will attempt to reply to email, but I **do not promise it**.
7. Try to give me enough information. Don’t send a message, “I am getting the error Invalid Syntax” and nothing else. Send me a screen dump of the error and what you did before the error and the file that caused that error. (Or whatever other information you can provide.)
8. If several people have the same question or problem I will “reply to all” and not to you individually.
9. Read MY email before you send me YOUR email. As to 8. above, I might have already replied to your concern.
10. You will always have at least two weeks to work on a project. IF AN ASSIGNMENT IS DUE ON WEDNESDAY, DON’T START WORKING ON IT ON SUNDAY MORNING AND EXPECT THAT YOU WILL GET EMAIL HELP IMMEDIATELY. START WORKING ON ASSIGNMENTS IMMEDIATELY.
1. I will not reply to a flood of messages over the weekend. See 5. above.

2. If the system crashes (and that happens) the system administrator is not obligated to fix it over the weekend. Sometimes he does, sometimes he does not. This will not be an excuse for missing a deadline.

Thank you for your cooperation.

---

**Late Work and Make-Up Exams**

Students will lose 8 points for 1 week. Projects handed in more than 1 week late might not get any points.

---

**Student Conduct**

The NJIT University code on academic integrity, found at [http://www.njit.edu/education/pdf/academic-integrity-code.pdf](http://www.njit.edu/education/pdf/academic-integrity-code.pdf), will be followed in all courses.

---

**Student with Disabilities Codes**

NJIT adheres to section 504 of the Rehabilitation Act (ADA) of 1990. 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](http://www.njit.edu/education/pdf/academic-integrity-code.pdf) 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](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: 1-888-789-0407 or [https://supportcenter.cmanet.com/NJIT](https://supportcenter.cmanet.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 anyone using a computer. All members of the university community are encouraged to review tips for password management and to change passwords regularly.

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
- 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 ext. 5112

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
CS 634 Data Mining

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

Prerequisite and Required Background
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/or (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 639 102 Elec Medical Records: Computer Implementation

(Medical Informatics)  
Spring 2016

**Syllabus**

<table>
<thead>
<tr>
<th>Course No.</th>
<th>CS 639 - 102</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
<td>Electronic Medical Records: Computer Implementation (Course will cover Medical Terminologies and Electronic Medical Records for Clinical/Healthcare IT).</td>
</tr>
<tr>
<td><strong>Day/Time</strong></td>
<td>Thursday, 6:00pm to 9:05pm</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>GITC, 4th Floor Room 4415</td>
</tr>
<tr>
<td><strong>Prerequisite(s)</strong></td>
<td>CS 631 or Undergraduate Database course or practical experience with a Database system, or <strong>obtain Permission from Instructor</strong>. 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><strong>Instructor</strong></td>
<td>Dr. Karen Hare</td>
</tr>
<tr>
<td></td>
<td>• GTIC CS Dept. 4th Floor, mailbox</td>
</tr>
<tr>
<td></td>
<td>• Dept. phone number: 973-596-3366</td>
</tr>
<tr>
<td></td>
<td>• Dept. fax number: 973-596-5777</td>
</tr>
<tr>
<td></td>
<td>• Email: <a href="mailto:karen.hare@njit.edu">karen.hare@njit.edu</a></td>
</tr>
<tr>
<td><strong>Instructor Office Hours</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Location: GITC CS Department, 4th Floor Room 4411</td>
</tr>
<tr>
<td></td>
<td>• Wednesday, 1:00 pm to 5:30pm By Appointment</td>
</tr>
<tr>
<td></td>
<td>• Thursday, 4:00pm to 5:30pm or By Appointment</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This course presents an introduction to Medical Informatics for Computer Science students by covering important topics from Biomedical and 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. 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 clinical software development.</td>
</tr>
</tbody>
</table>
CS 639 102 Elec Medical Records: Computer Implementation  
(Medical Informatics)  
Spring 2016  
Syllabus  
Dr. Karen Hare  
karen.hare@njit.edu

| Description (Cont’d) | Designed as an introduction 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 analyst, designers, developers, IT managers, decision makers', consumers and providers. Addressing the integration of Clinical healthcare applications with Computer Science skills like: programming, data mining, operating systems, networking protocols, security and privacy to develop and deploy innovations in the field of Healthcare Information Technology.  
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 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. Projects will require programming knowledge.  
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.** |
|---|---|
CS 635 - Computer Programming Languages - Spring 2016

Professor: Iulian Neamtiu  
email: ineamtiu@njit.edu  
Office hours: Fridays 9:30a.m.–11:30a.m.  
GITC 4108

Textbook (mandatory)  “Types and Programming Languages” aka TAPL  
Author: Benjamin C. Pierce  

Optional references  
"Introduction to Objective Caml" by Jason Hickey  
"Developing Applications with Objective Caml"

Prerequisites  
CS 505 and CS 510

Section  
002

Call #  
11474

Credits  
3

Class  
Fridays, 1:00p.m.–3:55p.m.  
KUPF 203

Holidays  
No class March 18, March 25

Catalog description  
The theory and design of computer language systems; the formal theory of syntax and language classification; a survey of procedure and problem-oriented computer programming languages, their syntax rules, data structures, and operations; control structures and the appropriate environments and methods of their use; a survey of translator types

Detailed schedule (subject to change)

<table>
<thead>
<tr>
<th>When</th>
<th>Topic</th>
<th>Reading (TAPL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 22</td>
<td>Introduction; OCaml</td>
<td></td>
</tr>
<tr>
<td>Jan 29</td>
<td>Untyped Arithmetic Expressions</td>
<td></td>
</tr>
<tr>
<td>Feb 5</td>
<td>Untyped Lambda Calculus</td>
<td>Ch. 2,3</td>
</tr>
<tr>
<td>Feb 12</td>
<td>Typed Arithmetic Expressions</td>
<td>Ch. 5</td>
</tr>
<tr>
<td>Feb 19</td>
<td>Simply Typed Lambda Calculus</td>
<td>Ch. 8</td>
</tr>
<tr>
<td>Feb 26</td>
<td>The Curry-Howard Isomorphism; Extensions to Simply Typed Lambda Calculus (I)</td>
<td>Ch. 9</td>
</tr>
<tr>
<td>March 4</td>
<td>Extensions to Simply Typed Lambda Calculus (II); Exceptions</td>
<td>Ch. 11</td>
</tr>
<tr>
<td>March 11</td>
<td>Midterm</td>
<td>Ch. 14</td>
</tr>
<tr>
<td>March 18</td>
<td>No class (Spring Break)</td>
<td>Ch. 13</td>
</tr>
<tr>
<td>March 25</td>
<td>No class (Good Friday)</td>
<td>Ch. 15, 16.1, 16.2</td>
</tr>
<tr>
<td>April 1</td>
<td>References</td>
<td>Ch. 18</td>
</tr>
<tr>
<td>April 8</td>
<td>Subtyping</td>
<td>Ch. 19</td>
</tr>
<tr>
<td>April 15</td>
<td>Imperative Objects</td>
<td>Ch. 23</td>
</tr>
<tr>
<td>April 22</td>
<td>Featherweight Java</td>
<td></td>
</tr>
<tr>
<td>April 29</td>
<td>Polymorphism: Universal Types</td>
<td></td>
</tr>
<tr>
<td>Tuesday May 3</td>
<td>Polymorphism: Existential Types; Review</td>
<td></td>
</tr>
</tbody>
</table>

Class score

<table>
<thead>
<tr>
<th>Percent Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quizzes</td>
</tr>
<tr>
<td>Assignments</td>
</tr>
<tr>
<td>Midterm</td>
</tr>
<tr>
<td>Final exam</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

Assuming your overall numerical grade is $x$, your letter grade is:

- $x < 63$ : 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

This scheme operates on real numbers hence there will be no rounding, up or down.
Quizzes
There will be several unannounced quizzes at the beginning of class. Quizzes are designed to test students' understanding of the material assigned in advance for that class (section or chapter in the textbook) or to reinforce material studied in the previous lecture.

Attendance, basic etiquette, and electronic device policy
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
Each assignment is individual. Please refrain from collaborating, or discussing solutions, with your classmates; or searching online for answers to questions.
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.

Make-up policy
No quiz make-ups.
No midterm make-up per se, but if for some good reason (e.g., illness), you can't take the midterm on the official date, your final will be comprehensive. You need advance permission from the instructor for this (i.e., prior to the midterm).
No make-up final.

Required background
Elementary set theory, proofs by induction.

Exposure to at least two programming languages, one OOP and the other preferably from another paradigm, e.g., functional.

Computer science principles: a procedure-oriented language such as C++; program design techniques; introductory data structures (linked lists, stacks, sets, trees, graphs); and algorithms (sorting, searching, etc.) and their analysis (CS 505).

Assembly language programming including basic machine organization, the structure of instruction sets, program linkage, macros and macro libraries (CS 510).
CS 639 102 Elec Medical Records: Computer Implementation  
(Medical Informatics)  Spring 2016
Syllabus  
Dr. Karen Hare  
karen.hare@njit.edu

<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. |

**NJIT Bookstore will have both books**

<table>
<thead>
<tr>
<th>Other Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exams:</strong></td>
</tr>
<tr>
<td>There will be one Midterm Exam, group project and one Final Exam. All exams and projects are face-to-face. Cellphones will not be allowed during the taking of exams.</td>
</tr>
</tbody>
</table>
Other Information (Cont’d)

**Homework:**
There will be 5 homework assignments involving working with EMR systems and terminology browsers. As well as, reading, writing article reviews, quizzes and Web research. All assignments must be in Word processing format. You will be graded on grammar, spelling, sentence structure and proper citations.

***There will be a penalty of 10 point per day for late assignments unless otherwise arranged with the instructors permission.***

***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 One.***

**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: 150
- Final Exam/Project: 150
- Homework 1: 50
- Homework 2: 50
- Homework 3: 50
- Homework 4: 50
- Homework 5: 50
- Extra Credit: 10

Total 560

If homework or exam points are miscalculated it must be addressed upon class review, otherwise, no point adjustments will be made! Especially at the end of the semester, prior to final grade submission.

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. Thus, your position in the curve and your class grade depend almost entirely on a combination of homework, projects, quizzes and exams. On the other hand, missing a single homework is highly likely to lower your grade at least one letter grade.
Learning Objectives

By the end of the class you will be able to:
• Use and understand Medcin software.
• Recognize key EHR/EMR terminology use in software applications.
• To understand how to integrate EHR/EMR and Computer Science concepts into real world IT applications.
• To understand user computer interfaces for clinical application design and development.

Plagiarism and Academic Integrity
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 [http://www.njit.edu/education/pdf/academic-integrity-code.pdf](http://www.njit.edu/education/pdf/academic-integrity-code.pdf) and understand 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/](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.

Cutting and pasting from books, articles, URL links and inserting into your homework falls in this category. Use appropriate citations, i.e., American Psychological Association (APA) format.

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](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:
| University Resources (Cont'd) | 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://list.njit.edu/support/contactus.php](https://list.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):**  
2. **Student Moodle Tutorials:**  
3. **Student Moodle FAQs:** [http://moodle.njit.edu/tutorials/students/faq.php](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.  
Counter Hacking Techniques

Course No.
CS 647

Sections
102

Title
Counter Hacking Techniques

Course Website
http://cs.njit.edu/~karvelas/CS-647-Spr16

Prerequisite(s)
One of CS-645, CS-646, CS-656 or ECE-638 or the approval of the instructor.

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
Wednesdays: 4.30 pm - 5.50 pm
    Thursdays: 5.10 pm - 5.50 pm
    Fridays:  5.10 pm - 5.50 pm

Description
This course covers advanced techniques that can be used for offensive or defensive goals in
networks, computer systems and applications. The course follows a “learning by doing”
teaching approach through extensive use of virtual machines with vulnerable operating
systems and applications. Topics covered include system memory organization, CPU
registers, assembly language fundamentals, GNU and Immunity debuggers, fuzzing based
security testing, development of local and remote Linux and Windows exploits, shellcode
development, bypassing memory protection techniques, ethical and legal issues in
information security.

Grading Scheme
Two Quizzes:  10% each
Two lab assignments: 10% each
Midterm:  28%
Final:  32%
Topics
1. Computer memory organization, CPU registers, assembly language fundamentals
2. Debugging with GDB
3. Stack-based buffer overflows
4. Heap and BSS-based overflows
5. Format-string vulnerabilities
6. Writing Linux shellcode
7. Remote Linux exploitation
8. Immunity debugger, fuzzing based security testing
9. Windows system exploitation
10. Jumping to shellcode techniques
11. SEH based exploits
12. Memory protection techniques and how to bypass them
13. Ethical and legal issues in information security

Disclaimer:
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.

Learning Goals and Measurable Learning Outcomes
After completing the course the students will have hands on experience in:
1. Testing for and discovering vulnerabilities in applications
2. Developing exploits for vulnerable local/remote Linux applications
3. Developing exploits for local/remote Windows applications
4. Recommending appropriate mitigation countermeasures.

Text Book(s)
No specific textbook will be used for this course. The material will be based on class notes, conference, and journal papers posted on the class website.

Time & Place
Mondays, 6.00 pm – 9.05 pm, CKB 206

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
002

Title
Computer Networks - Architectures, Protocols and Standards

Course Website
http://cs.njit.edu/~karvelas/CS-652-Spr16-Mrrn

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
Wednesdays: 4.30 pm - 5.50 pm
Thursdays: 5.10 pm - 5.50 pm
Fridays: 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, 1.00 pm – 4.05 pm; CKB 206

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.
Faculty Contact Information

Instructor: Dr. Grace Guiling Wang, Associate Professor of Computer Science
Email: guiling.wang@njit.edu

Teacherv's Assistant: Yuan Lu
Email: y1768@njit.edu

Office Hours: Office hours will be held through WebEx. The detailed time will be posted in the "Instructor Announcement" discussion forum in the first week of the class.

Course Description

This course studies the architecture and protocols of modern computer networks. Topics to be covered include: addressing, performance measurement metrics, application-layer protocols, transport-layer protocols, networking-layer protocols, link-layer protocols, and wireless and mobile networking. 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.

Textbook and Materials


Prerequisite

Decent programming skills are required. The class has a semester-long programming project and the professor will not debug this assignment for you. CS356 or equivalent courses are suggested.

Course Outcomes

- Deep understanding of modern computer network architecture and protocols.
• Broad knowledge on start-of-the-art network concepts.
• Hands-on experience in designing and coding network-related mobile application.
• Improved presentation skills.

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>&lt;= 25%</td>
</tr>
<tr>
<td>B+</td>
<td>3.5</td>
<td>&lt;= 25%</td>
</tr>
<tr>
<td>B</td>
<td>3.0</td>
<td>&lt;= 25%</td>
</tr>
<tr>
<td>C+</td>
<td>2.5</td>
<td>&lt;= 15%</td>
</tr>
<tr>
<td>C</td>
<td>2.0</td>
<td>&lt;= 10%</td>
</tr>
<tr>
<td>F</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Reminder: At the discretion of the instructor, the grading may be done on a curve.

Grading Categories

<table>
<thead>
<tr>
<th>Categories</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiz</td>
<td>8%</td>
</tr>
<tr>
<td>Research / Presentation Assignment</td>
<td>20%</td>
</tr>
<tr>
<td>Programming Assignment</td>
<td>20%</td>
</tr>
<tr>
<td>Mid-Term Exam</td>
<td>20%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>32%</td>
</tr>
</tbody>
</table>

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

Quiz Dates

<table>
<thead>
<tr>
<th>Quiz</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 03: Quiz 01</td>
<td>February 7</td>
</tr>
<tr>
<td>Module 05: Quiz 02</td>
<td>February 21</td>
</tr>
<tr>
<td>Module 10: Quiz 03</td>
<td>April 3</td>
</tr>
<tr>
<td>Module 13: Quiz 04</td>
<td>April 24</td>
</tr>
</tbody>
</table>

Group Assignments
Both programming assignment and research assignment are done in a group and in the same group. Continue reading on Team Formation and those expectations.

Objectives of team work

The objective of teamwork is two-fold: (1) Communication is an art. Through teamwork, you are expected to learn and improve on how to collaborate and communicate with other people. (2) Teammates are expected to learn from each other. Each individual has his/her own gift. Learn from your peers. Also, one person cannot do everything, and you are expected to learn how to leverage others’ resources and help.

Especially for an online course, teamwork is very important and more than necessary because you do not go to the classroom for lectures together. Without required teamwork, you may not even know your classmates at the end of the semester. Learning from each other and making friends at graduate school is very helpful to future career development.

Team formation requirements/suggestions

- [Requirement] A group is composed of four members. Considering that the number of enrolled students may not be a multiple of four, two groups can have five members or three members.
- [Rule] You need to build a team by the end of the first week.
- [Rule] After week two, you cannot change groups anymore except for some extreme situation, e.g., all your partners drop the class. That your partner does not work or that you do not like them is not a legitimate reason to change partners. So BE CAREFUL when forming a team.
- [Rule] Once you have decided on which group you belong to, put your name beside your other group members on this spreadsheet: Groups. The group member who signs up first under a blank group on the spreadsheet decides the group number. For example, if John Smith sees that group 2 is blank on the spreadsheet; then he can start group 2, and all other group members who are in a group with John are now in group 2. Also, immediately respond to the thread that corresponds to your group number. For example, if John Smith signs up for group 2 on the spreadsheet, he should respond to the group 2 thread saying something like, "John Smith joining the group!"
- [Administration] Use the thread below that corresponds to your group to facilitate conversation on your group assignments. Once all groups are finalized we will restrict the settings so that this thread can only be seen by your other group members. Once restricted, posts in your private group thread will only be visible to you, your other group members, and the teaching staff. Not other groups.
- [Administration] Once you form a group, you can start bidding the project topics that your group plans to work on.
- [Suggestion] The best strategy to form a team is to incorporate different skills, so that (1) group members can learn from each other; (2) the group possesses the whole skill set necessary to accomplish the project including programming, information searching, presentation, and many more.
- [Suggestion] Be careful when forming a team. Reach out to all your other classmates, if possible, and find out the right partners to work with.

About Grading

A project has a grade. All the group members will have the same grade. No request for a higher grade than team members will be granted for any reason.

About fairness

Note that you may be the one who does most of the work and your teammate does not
finish his/her tiny parts. You may feel it is unfair. However, you need to know similar or worse things happen more often in the real work place. You need to learn how to handle this. First of all, if you are the one who does most of the job, you learn the most from the project and no one can steal gained knowledge and experience from you. You actually have the opportunity to practice while your teammates give that up. Secondly, effective communication with others is not easy. How do you make others do their work? How do you express your feelings diplomatically? You need to learn to handle different situations and different kinds of people.

About backup plan

The whole group will have the same grade. If your group determines to split the project and each one works on a piece, it is possible that one member does not do his/her job and potentially the grade of those who have finished their parts are affected equally. Therefore, from day one, there should be a backup plan or some monitoring mechanism in case a team member drops the class or just does not finish his/her parts.

Programming Assignment

- The project is about developing a smart phone application.
- The platform can be Android phone or iOS.
- Different groups will work on different projects or on different platforms. The professor will provide a list of projects and groups will bid on the project/platform.
- All students, including the presenting group, will participate in the Peer Review Questionnaire.

More details about the programming assignment can be found in a separate description. Note: The professor will NOT teach you how to develop a mobile application, NOR debug it for you. Instead, the professor will post useful online material so you can learn by yourself with your team members. Smart phone app development skills are in great demand in today's job market. As a graduate student with decent programming skills (our course prerequisite), you definitely would be able to do it after spending time. Hands-on experience is important. We need to obtain the most useful network-related hands-on experience.

Research Assignment

- The research assignment is for you to study by yourself and with your team members some hot networking topics or new networking technologies.
- The professor will provide a list of research assignment topics. Each group will work on a topic. Other groups will learn from this group about the topic. The presenting group needs to answer the questions raised by their classmates.
- All students, including the presenting group, will participate in the Peer Review Questionnaire.

More details about the research assignment can be found in a separate description.

Quizzes

There will be four quizzes in total. You have a 48-hour-long window to take the 25 min-quiz online. Quiz date cannot be rescheduled. Any request to change the quiz date will not be considered nor replied to. However, you can choose to drop the grade of one quiz. By default, we will drop the quiz with the lowest score and the final quiz score will be the average of the highest three scores unless you indicate...
## Homework Exercises

Homework exercises will be given almost every week and the solutions will be provided at the same time. You need to work on the exercises, check with the solutions, and ensure you know how to solve them.

There is no need to submit your homework exercises. They will not be counted into the final grade.

## Time Commitment

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 19 – January 24</td>
<td>Introduction / Protocol Layer and Service Models</td>
</tr>
<tr>
<td>January 25 – January 31</td>
<td>Network Performance: Delay and Loss/ Addresses</td>
</tr>
<tr>
<td>February 1 – February 7</td>
<td>Application Layer Architecture/ FTP and E-mail/P2P Applications</td>
</tr>
<tr>
<td>February 8 – February 14</td>
<td>The Web, HTTP, and DNS</td>
</tr>
<tr>
<td>February 15 – February 21</td>
<td>Web Searching/ Intro to Transport Layer/ UDP</td>
</tr>
<tr>
<td>February 22 – February 28</td>
<td>Principles of Reliable Data Transfer/ TCP Reliability</td>
</tr>
<tr>
<td>February 29 – March 6</td>
<td>TCP Congestion Control</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td></td>
</tr>
<tr>
<td>March 7 – March 13</td>
<td>Router/ IP</td>
</tr>
<tr>
<td>March 21 – March 27</td>
<td>Routing Algorithms</td>
</tr>
<tr>
<td></td>
<td>Introduction to Link Layer/ MAC Protocol</td>
</tr>
</tbody>
</table>
Module 11
April 4 – April 10
Switched Local Network/ Overview of the Five Layers

Module 12
April 11 – April 17
Features of Wireless Links/ WiFi

Module 13
April 18 – April 24
Cellular Network/ Manage Mobility

Module 14
April 25 – May 1
Group Project Overviews

Final Exam

Disclaimer
The module 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 modules.

Policies about Emails and Questions

- All the academic questions should be posted through Moodle on the corresponding discussion forums and the professor/TA will answer there. In this way, students can learn from each other’s questions. For such a post, use [Questions to Professor/TA] in the subject. Before posting a question, first check whether a similar question has been posted already and answered. **Such questions emailed to the professor/TA will not be answered.** For example, if you have a question about how congestion control works, post on the forum, asking by email will not be replied to. If you have a question about course administrative details, e.g., the TA’s office, check corresponding documents (in this case, the syllabus) first. If no answer, post online.

- Only if you have some personal issue, contact the professor/TA by email. For example, if all your group members drop the class and you need help.

- If you have comments and suggestions about how to improve the class, please email the professor/TA, instead of posting on Moodle. For example, you believe that one programming topic is too hard and the professor should change a topic. The professor appreciates the comments and will consider adjusting the course material for the coming semesters, but the requirement clearly stated from the beginning will be strictly followed.

- Questions posted online will be answered by 11:59PM on the next business day. For example, if you post a question on Friday at 10AM, the question will be answered by Monday at 11:59PM. If you post a question on Monday at 10PM, the question will be answered by Tuesday at 11:59PM. The only exception is when the professor is out of town for academic conferences/meetings and has no Moodle access, and the TA cannot answer the question. In this case, the Professor will answer the question within one business day after she comes back.

Late Work and Make-Up Exams

A make-up exam request and request to use longer time to finish the exam may be taken only after providing written documentation from the Dean of Students. The Dean
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 an 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/academics/integrity.php, will be followed in all courses.

Student with Disabilities Codes

NJIT adheres to section 504 of the Rehabilitation Act (ADA) of 1990. 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) 696-3420. 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: 1-888-789-0407 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
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
CS656:002 Internetworking & Higher Layer Protocols

Class time  11:30-2:25PM Thursday  
Classroom  Kupfrian Hall 208  

Professor  Dr. Grace Guiling Wang  
Email:  gwang@njit.edu  
Office :  GITC 4309  
Office hour:  10:00-11:00AM Wednesday or by appointment  

TA  Yuan Lu  
Email:  yl768@njit.edu  
Office:  GITC 4325  
Office hour:  4:00-5:00PM Thursday & Friday or by appointment  

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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></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.
New Jersey Institute of Technology
Department of Computer Science

CS670 - Artificial Intelligence - Spring'2016

Tuesday 6:00 - 9:05 PM, CKB 219

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: Monday 1:25-2:25PM, Tuesday 3:30-5:30PM, or by appointment

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
   - Artificial Neural Networks (Perceptrons, RBF)
9. Learning - Unsupervised Learning
   o Principal Component Analysis
   o Applications: Compression, Feature Representation

10. Learning - Supervised Learning
    o Discriminant Analysis
    o Applications: Feature Extraction for Classification

11. Learning - Probabilistic Models
    o Statistical Learning Theory (STL)
    o Structural Risk Minimization (SRM)
    o Support Vector Machines (SVM)

12. Learning - Other Popular Models (optional)
    o The EM Algorithm
    o Bayesian Learning
    o Genetic Algorithms
    o Reinforcement Learning

13. Perception - Search in Spatial Domain and Frequency Domain (optional)
    o FFT, Lowpass and Highpass Filtering, Convolution Theorem
    o Edge Detection, Line and Curve Detection (Hough Transform)
    o Pictorial Information Search using Geometric or Frequency Features

    o Sensors and Vision
    o Path Planning
    o 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:
  o J.R. Fisher, The Prolog Tutorial
- Lisp:
CS680 Linux Kernel Programming - Course Syllabus

- Class Web page: http://cs.njit.edu/~sohna/cs680
- Instructor: Andrew Sohn, GTIC 4209, (973)596-2315, email: sohn at cs dot njit dot edu
- Office Hours: MW 11:30am-1pm
- No Teaching Assistant: NJIT policy; grader TBA
- Class time and location: Tue, 10 am - 1 pm, CUL Lec 3
  http://www.njit.edu/registrar/schedules/courses/spring/2016S.CS.html
- Kernel version 4.4: download the latest version at kernel.org
- Books recommended:
- Grading: 10-12 kernel programming assignments (20%), Test 1 (20%), Test 2 (20%), final exam (40%). In-class demonstration using your laptop is required for lxr – Linux cross reference. Homework will not be accepted after the due date. Submit it on time. Do not email your homework. Emailed homework will be discarded.
- NJIT policy on missed exams: There will be no make up exam(s). You must plan your semester accordingly, especially if you work full time. If you should miss the exam(s) due to emergency, go to the Dean of students and explain your situation as to why you had to miss. I will accept Dean’s memo. This is the NJIT policy for missed exams. No other policy will be applied.
- The NJIT Honor Code: See Academic Integrity

Schedule by Week

1. LAMP, virtualization, datacenter 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 reference) on your laptop and in-class demonstration, compiling the kernel, Module programming, Booting - machine BIOS, disk MBR, Grub Linux loader, preliminary setup (setup0, startup_32) 1 and 2
3. Booting continues, Overview of kernel startup and initialization (start_kernel)
   Memory - overview, segmentation
4. Memory - mapping, paging (get_free_pages),
5. Test 1: 10-11:30am, Tue, 2/16/2016 (week5)
   Memory - caching (kmalloc) and process address space (vmalloc)
6. Process - thread union, task and thread struct, PID0 (swapper)
7. Process - PID1 (init), PID2 (kthreadd), PID3 (softirqd), PID4 (migrationd)
8. Process - process scheduling and process switching
   Interrupts - exceptions (traps)
9. Interrupts - hard interrupts do_IRQ()
10. Test 2: 10-11:30am, Tue, 3/29/2016 (week10)
   Interrupts - soft interrupts do_softirq(this), ksoftirqd, timer interrupts
11. File system - virtual file system, registering, mounting, Block IO,
12. File system - IO scheduler, device driver, vfs_read, Ext4 example
13. Networking - receiving packets: NIC, ISR, Softirq, IP, TCP, Inet, BSD, User
14. Networking - sending packets: User, BSD, Inet, TCP, IP, Softirq, Qdisc, ISR, NIC
15. Final exam: See the registrar's page: http://www.njit.edu/registrar
Spring 2016 CS680 Linux Kernel Programming - Course Syllabus
Schedule by Week

1. LAMP, virtualization, datacenter 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) Memory - overview, segmentation

4. Memory - mapping, paging (get_free_pages),

5. Test 1: 10-11:30am, Tue, 2/16/2016 (week5)
   Memory - caching (kmalloc) and process address space (vmalloc)

6. Process - thread union, task and thread struct, PID0 (swapper)

7. Process - PID1 (init), PID2 (kthreadd), PID3 (softirqd), PID4 (migrationd)

8. Process - process scheduling and process switching
   Interrupts - exceptions (traps)

9. Interrupts - hard interrupts do_IRQ()

10. Test 2: 10-11:30am, Tue, 3/29/2016 (week10)
    Interrupts - soft interrupts do_softirq(), ksoftirqd, timer interrupts

11. File system - virtual file system, registering, mounting, Block IO,

12. File system - IO scheduler, device driver, vfs_read(), Ext4 example

13. Networking - receiving packets: NIC, ISR, Softirq, IP, TCP, Inet, BSD, User

14. Networking - sending packets: User, BSD, Inet, TCP, IP, Softirq, Qdisc, ISR, NIC

15. Final exam: See the registrar's page: http://www.njit.edu/registrar
CS 685: Software Architecture
Syllabus, Spring 2016

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

Fall 2016 Office Hours: Tuesday 1:00-2:30, Friday 2:30-4:00, or by appointment

**CS 685 – Software Architecture**
The software architecture defines the structure and interactions of software modules. This course provides a working knowledge of the terms, principles and methods of software architecture and module design. It explains the constraints on the design and the properties of capacity, response time, and consistency. The "4+1" architecture model is taught with architectural styles, interface isolation, decoupling, reuse, agile design with software patterns, data structures, queueing effects, design simplification and refactoring. The non-functional requirements of reliability, performance and power consumption, component based design and good industry practices for documenting and managing the architectural process are taught.

Please include CS685 and 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.

**Required Background:** http://web.njit.edu/~gwryan/CS685

**Course Webpage:** http://web.njit.edu/~gwryan/CS685
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

**Textbook:**
Grading

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>50%</td>
</tr>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Final</td>
<td>30%</td>
</tr>
</tbody>
</table>

Project:

Half of the grade for the course is a project with five deliverables over the course of the semester. The project is done in teams with each team responsible for a different product family, and with each team member responsible for a different product in that family. The project will thus have portions of the grade that are individual and portions of the grade that are shared with the team.

The project consists of five deliverables/milestones:

1. A report on downloading, installing and using each product (5%)
2. A report on the architecture of each product (20%)
3. A report on a comparison of the products in the product family (30%)
4. An in-class presentation on the architectures and the comparison (25%)
5. A report on a reference architecture for the product family (20%)

Each paper must be submitted as a PDF. The paper should be single spaced 11 point Times New Roman, or a monospaced font such as Courier for any source code or command lines used in the paper. Your margins should be .75 inches on all four sides.

The first page of the report should begin with:

- CS 685 Spring 2016
- Group name or product family name
- Authors names and email addresses
- Milestone number for the report
- Title for the report

Be sure to cite references in your paper.
Ethical Conduct

Cheating during in-class tests or take-home examinations or homework is, of course, illegal and immoral. You may discuss problems with each other, but you may NOT copy information from anybody or anywhere without attribution. Giving your work to someone else to copy from is just as much cheating as copying someone else’s work.

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](http://www.njit.edu/academics/pdf/academic-integrity-code.pdf)

Course Outline:

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Text</th>
<th>Milestone Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>22-Jan</td>
<td>Introduction and Terminology</td>
<td>Ch 1, 2</td>
<td></td>
</tr>
<tr>
<td>29-Jan</td>
<td>Module Structures</td>
<td>Ch 3, 4, 13</td>
<td>Milestone 0 due</td>
</tr>
<tr>
<td>5-Feb</td>
<td>Component &amp; Connector Structures</td>
<td>Ch 13</td>
<td></td>
</tr>
<tr>
<td>12-Feb</td>
<td>Component &amp; Connector Structures</td>
<td>Ch 13</td>
<td>Milestone 1 due</td>
</tr>
<tr>
<td>19-Feb</td>
<td>Component &amp; Connector Structures</td>
<td>Ch 13</td>
<td></td>
</tr>
<tr>
<td>26-Feb</td>
<td>Allocation Structures</td>
<td>Ch 13</td>
<td></td>
</tr>
<tr>
<td>4-Mar</td>
<td>Midterm</td>
<td></td>
<td>Milestone 2 due</td>
</tr>
<tr>
<td>11-Mar</td>
<td>Working as an Architect</td>
<td>Ch 14, 16-18</td>
<td></td>
</tr>
<tr>
<td>1-Apr</td>
<td>Attributes</td>
<td>Ch 5-8</td>
<td></td>
</tr>
<tr>
<td>8-Apr</td>
<td>Attributes</td>
<td>Ch 9-12</td>
<td>Milestone 3 due</td>
</tr>
<tr>
<td>15-Apr</td>
<td>Web Architectures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22-Apr</td>
<td>Samples and Studies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29-Apr</td>
<td>Project presentations (Milestone 4)</td>
<td></td>
<td>Milestone 4 due</td>
</tr>
<tr>
<td>3-May</td>
<td>Project presentations (Milestone 4)</td>
<td></td>
<td>Milestone 5 due</td>
</tr>
</tbody>
</table>
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

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 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.
6 GRADING PLAN

Final grades are weighted as follows:

1. Homework: 30%
2. Project: 30%
3. Mid-term Exam: 20%
4. 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 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.

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.

7 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.
## 8 TENTATIVE SCHEDULE

<table>
<thead>
<tr>
<th>Module #</th>
<th>Theme</th>
<th>Lecture Hour 1</th>
<th>Lecture Hour 2</th>
<th>Lecture Hour 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Networks</td>
<td>Introductions &amp; Policies</td>
<td>Network Stack</td>
<td>Protocol Families</td>
</tr>
<tr>
<td>2</td>
<td>Malware</td>
<td>Common vulnerabilities</td>
<td>Malware</td>
<td>Counter-measures</td>
</tr>
<tr>
<td>3</td>
<td>Measurements</td>
<td>Network Measurements</td>
<td>Tools 1</td>
<td>Tools 2</td>
</tr>
<tr>
<td>4</td>
<td>Circumvention</td>
<td>Firewalls</td>
<td>Tor and friends</td>
<td>Browser Fingerprinting</td>
</tr>
<tr>
<td>5</td>
<td>Wireless</td>
<td>Wireless protocols</td>
<td>WEP and WPA</td>
<td>Cellular Data Services</td>
</tr>
<tr>
<td>6</td>
<td>Web</td>
<td>DNS</td>
<td>SSL/TLS</td>
<td>Exam Topics</td>
</tr>
<tr>
<td>7</td>
<td>Exam</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Crypto</td>
<td>Encryption Standards</td>
<td>Public Key Crypto</td>
<td>AES and friends</td>
</tr>
<tr>
<td>9</td>
<td>Key Management</td>
<td>PKI</td>
<td>Diffie-Hellman</td>
<td>Chap / Kerberos</td>
</tr>
<tr>
<td>10</td>
<td>Embedded Systems</td>
<td>Embedded Networking</td>
<td>CAN Bus</td>
<td>Automotive Attack Surfaces</td>
</tr>
<tr>
<td>11</td>
<td>TBD – selected by students</td>
<td>Chat</td>
<td>XMPP</td>
<td>OTR</td>
</tr>
<tr>
<td>12</td>
<td>TBD – selected by students</td>
<td>Physical Access Control</td>
<td>Physical Security Networks</td>
<td>Biometrics</td>
</tr>
<tr>
<td>13</td>
<td>Review and Discuss Projects</td>
<td>ARP</td>
<td>SOAP</td>
<td>Project Discussion</td>
</tr>
<tr>
<td>14</td>
<td>Exam</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Network Management and Security

Course No.
CS 696

Sections
102

Title
Network Management and Security

Course Website
http://www.cs.njit.edu/~karvelas/CS-696-Spr16

Prerequisite(s)
CS-652 or CS-656 or equivalent

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
Wednesdays: 4.30 pm - 5.50 pm
Thursdays: 5.10 pm - 5.50 pm
Fridays: 5.10 pm - 5.50 pm

Description
Network management and security are essential factors for the reliable, efficient, and secure operation of networks. As businesses become increasingly dependent on networking services, keeping these services running and secure becomes synonymous with keeping the business running. This course provides a thorough introduction to network management technologies and standards as well as to wide variety of techniques for evaluating, monitoring, and defending the security of computer networks and systems. The course covers the SNMP family of standards including SNMP, SNMPv2, SNMPv3, and RMON (Remote Monitoring), techniques for evaluating the security of computer systems, how attackers can compromise vulnerable systems, and how to monitor and defend the security of both computer networks and systems.

Grading Scheme
Two Quizzes: 10% each
Two lab assignments: 10% each
Midterm: 28%
Final: 32%
Topics
1. Introduction to Network Management
2. Overview of Network Technologies and TCP/IP Protocols
3. Network Management Standards, Models and Language
4. SNMPv1 Organization and Information Models
5. SNMPv1 Communication and Functional Models
6. Elements of SNMPv2 and SNMPv3
7. Remote Monitoring (RMON)
8. Introduction to Security and Cyber Security Threats
10. Techniques for Compromising Vulnerable systems
11. Advanced Post Exploitation Techniques
12. Network Security Monitoring

Learning Goals and Measurable Learning Outcomes
1. In depth understanding of IP addresses, subnetting, IP, TCP, UDP, DHCP, DNS protocols
2. Hands on experience with Wireshark and tcpdump packet analyzers.
3. In depth understanding of Network Management models and language.
4. In depth knowledge of SNMPv1 and its difference with SNMPv2 and SNMPv3.
5. In depth understanding of non-intrusive reconnaissance techniques using scripts and various reconnaissance tools such whois, nslookup, host, harvester, and email-collector.
6. In depth understanding of host, port, and services scanning techniques using Nmap scanner.
7. In depth understanding of host vulnerability analysis using Nessus.
8. In depth understanding of host exploitation techniques using the Metasploit framework.
9. In depth understanding of host post exploitation techniques using the Metasploit framework.
10. In depth understanding of network attacks using Nemesis and Ettercap tools.
11. Understanding of the fundamental security risk management concepts and corresponding terminology.

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

Time & Place
Fridays, 6.00 pm – 9.05 pm; CKB 317

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
102

Title
Computer Networks - Architectures, Protocols and Standards

Course Website
http://cs.njit.edu/~karvelas/CS-652-Spr16-Evn

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
Wednesdays: 4.30 pm - 5.50 pm
Thursdays: 5.10 pm - 5.50 pm
Fridays: 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)

Time & Place
Wednesdays, 6.00 pm – 9.05 pm; CKB 214

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.
CS 698: Special Topics in Big Data

General Information:

<table>
<thead>
<tr>
<th>Instructor</th>
<th>Chase Wu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office Room</td>
<td>GITC 4107</td>
</tr>
<tr>
<td>Office Hours</td>
<td>T, 4:00 pm - 6:00 pm, or by appointment with the instructor</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:chase.wu@njit.edu">chase.wu@njit.edu</a></td>
</tr>
<tr>
<td>Phone</td>
<td>973-642-4579</td>
</tr>
<tr>
<td>Lecture Time</td>
<td>M, 6:00 pm - 9:05 pm</td>
</tr>
<tr>
<td>Lecture Place</td>
<td>Kupfrian Hall 203</td>
</tr>
<tr>
<td>Teaching Assistant</td>
<td>TBA</td>
</tr>
</tbody>
</table>

Course Materials:

- Syllabus
- Chapter 1. Introduction
- Chapter 2. Computing Trends
- Chapter 3. Overview of Big Data Analytics
- Chapter 4. Big Data Analytics Platforms
- Chapter 5. Big Data Storage and Processing
- Chapter 6. Big Data Analytics Algorithms
- Chapter 7. Big Data Visualization
- Chapter 8. Big Data Workflow Management and Optimization

Late policy: No late work will be accepted unless prior arrangements are made with the instructor.

Homework/Project assignments:

Homework 1 (20 points), assigned on Monday, xx/xx/2016, due on Monday (11:59 pm), xx/xx/2016
Homework 2 (10 points), assigned on Monday, xx/xx/2016, due on Monday (11:59 pm), xx/xx/2016
Homework 3 (30 points), assigned on Monday, xx/xx/2016, due on Monday (11:59 pm), xx/xx/2016
Homework 4 (60 points), assigned on Monday, xx/xx/2016, due on Monday (11:59 pm), xx/xx/2016
Project (100 points), assigned on Wednesday, xx/xx/2016, due on Monday (11:59 pm), xx/xx/2016

Good Friday - No Classes: 03/25/2016, Friday
Last Day to Withdraw: 03/28/2016, Monday
Last Day of Classes: 05/03/2016, Tuesday
Final Exam: TBA

Final Grades Due: 05/17/2016, Tuesday
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
Department office: GITC 4400
Department phone: 973-596-5778
Course website:

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.

Required Background
Programming Skills
- C/C++ or Java in Linux
Prerequisite Courses
- 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>Attendance</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<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 on 4 V’s of Big Data Applications</td>
</tr>
</tbody>
</table>
| 3    | • Trends of Computing for Big Data  
  • High-performance Computing (Supercomputers and Clusters)  
  • Grid Computing  
  • Cloud Computing  
  • Mobile Computing |
| 4, 5 | • Big Data Overview  
  • Drivers of Big Data  
  • Big Data Attributes  
  • Data Structures  
  • Big Data Ecosystem  
  • Examples of Data Analytics |
| 6, 7 | • Big Data Tools, Techniques, and Systems  
  • Exascale Computing  
  • HDFS, HBase, and NoSQL (Document Store, Graph DB, etc.)  
  • MapReduce, Spark, Oozie, Tez, Hive, Pig, etc.  
  • Hadoop 1 and Hadoop 2 (YARN) |
| 8, 9 | • Advanced Analytical Theory and Methods  
  • Hadoop/Mahout  
  • Recommendation  
  • Clustering  
  • Classification  
  • Regression |
| 10, 11 | • Review and Midterm Exam |
| 12, 13, 14 | • Advanced Topics  
  • Big Data Visualization  
  • High-performance Networking for Big Data Movement  
  • Big Data Scientific Workflow Management and Optimization |
| 15 | • Review |