CSCI 1133 - Introduction to Computing and Programming Concepts  
Spring 2016 Syllabus

Course Description:  
CSCI 1133 offers an introduction to the fundamental principles of Computer Science and programming with an emphasis on problem solving and representing/computing with data. The course serves as an introduction for computer science majors, and it functions as a prerequisite for most of the higher-level computer science courses. Students will use the programming language Python to implement solutions to a broad spectrum of interesting and motivating computational problems. Specifically, the course will cover:

- An introduction to imperative and object-oriented programming paradigms
- The concept and properties of algorithms
- Problem solving skills and the role of abstraction and modularity in the problem-solving process
- Fundamental design concepts and principles (abstraction, program decomposition, encapsulation and information hiding, separation of behavior and implementation, etc.)
- Fundamental problem solving approaches (recursion, iteration, bisection, exhaustive enumeration, etc.)
- Fundamental data types and structures (numbers, strings, lists, associative structures, etc.)
- Fundamental search methods (linear search, binary search, etc.)
- Development and representation of abstract data types using an object-oriented model

Prerequisites  
CSCI 1133 does not require or assume any previous programming knowledge, however in this section, most people have had some programming experience (even if only a few hours playing around with a different programming language). Prior exposure to a first semester Calculus course (or concurrent registration) is a formal prerequisite. Some material from Calculus I (differentiating polynomials and the like) may be employed in lectures and lab assignments; moreover, the mathematical and logical reasoning skills developed in Calculus I will play a significant role in this class. It is important that you have the mathematical maturity to both reason with abstractions and generalize abstract concepts from specific problem statements.

Textbook  

Class Website  
All announcements and class information are provided via our class Moodle webpage. It's very important to stay abreast of the information on the site because it often changes. The reading and assignment schedule is actively updated and should be checked frequently. To access the class Moodle page use the following URL: www.moodle.umn.edu
Schedule
The class schedule is maintained on Moodle and subject to change without notice. You should check the online schedule frequently for announcements and updates.

Office Hours
Office hours are available on a walk-in basis during numerous times throughout the week. The schedule is maintained on Moodle.

What you should expect to learn from this course
CSCI 1133 will exercise and further develop your general analytical thinking and problem solving skills. Upon successful completion of the course you should be able to:

- Understand the notion of a computational process and express computational processes using Python programs.
- Employ effective strategies for the solution of computational problems using appropriate methods and data representations.
- Select appropriate data structures and algorithms to solve a variety of problems using imperative and object-oriented programming techniques.
- Understand the basic principles of program design, using abstractions and problem decomposition to hide implementation details and manage the intellectual complexity of the problem.
- Identify, explain, and use fundamental data types introduced in this class, including numbers, strings, lists, associative structures, Abstract Data Types, etc.
- Trace the execution of a variety of code segments and explain their computations.
- Test and debug programs to be sure they are producing correct results.
- Understand fundamental computer science concepts such as efficiency and computational complexity.
- Understand what an abstract data type is and how to create new abstract data types with loose coupling between components/behaviors.
- Understand recursion, solve problems recursively, and understand the relationship between recursion and iteration.

Course Content
Weekly class meetings for this course consist of three 50-minute lectures and one 2-hour lab. In addition you should expect to spend 6-8 hours per week outside of class on independent programming projects and reading assignments.

Lectures
Course material is presented during lectures that is not covered in the textbook and not generally made available outside of the lecture. Since you are responsible for this material it is very important to attend all lecture sessions. Some lecture examples may be posted online, however lecture notes are not posted and are not available from the instructor. If a lecture is missed, it is your responsibility to arrange for notes from a classmate.

Weekly Labs
There are a total of 15 weekly computing labs. The labs include a variety of exercises that introduce key concepts through hands-on programming exercises in a mentored and collaborative setting. The exercises are designed to expose you to essential computer science concepts and language features that you will use to complete the homework assignments and which you will be subsequently tested. The lab exercises also expose you to interesting problems and provide a sense of the breadth of
computing applications. The core part of each lab is intended to be successfully completed by all students during a single 2-hour lab session. Most of the labs contain an optional part that explores the concepts at a deeper level; after you've completed the required core, you can go onto the optional part to enhance your learning and understanding.

Lab attendance is mandatory, as the weekly lab exercise must be completed and checked before the end of the lab session. Labs may be completed with a partner of your choosing. Lab exercises are graded on a "pass/fail" basis. Note that you must pass a minimum of 13 of the 15 lab exercises to receive a passing grade for this class (this includes excused absences!).

This is a course where you learn by doing. Lab exercises are designed to help you gain familiarity with new material in a supervised and freely collaborative setting where you may seek help from the TAs, other students, or any other reference material provided that you do not directly copy any other person's work or include any portion of someone else's work in your solution.

**Individual Programming Assignments**

30% of your final grade consists of weekly independent programming assignments that vary in complexity and difficulty. Unlike lab exercises, programming assignments are not collaborative efforts and must be completed on your own without the assistance of anyone other than the course instructor or TAs. Each programming assignment consists of one or more computational problems that must be solved and programmed in the Python programming language. On average, you should expect to spend 4-8 hours on each homework assignment.

Programming assignments are graded on correctness, completeness, and style. Correctness and completeness refer to how well the program works. Style includes good design, readability (indentations, descriptive names for variables and procedures, appropriate use of blank spaces, etc.), and useful comments.

Note that individual programming assignments are submitted electronically and are automatically scanned for plagiarism violations. It is essential that you carefully follow the submission instructions precisely or your assignment will not be accepted for grading. Failing to properly submit an assignment will result in a score of zero.

**Examinations**

There are two written midterm exams (20% each) and a comprehensive written final exam (30%). The exams are not team efforts; all exam responses must be your own. No electronic devices are allowed during exams. This includes calculators, cell phones, laptops, MP3 players, etc. All exams for this class are closed-book with restricted notes. Restricted means that you may bring a single, 8.5x11 sheet of paper with handwritten notes only (one or both sides). No printouts or xerographic copies are allowed.

**In-class Quizzes**

Short unannounced quizzes may be periodically administered during lectures without prior notice. These quizzes will generally cover material from reading assignments, prior lectures and homework assignments. In-class quiz scores will be included with the programming assignment portion of your semester grade.

**Grading**

Your final grade will be based on the following weights:

- 30% Individual programming assignments and unannounced in-class quizzes
- 40% 2 Written midterm exams @ 20% each
- 30% Comprehensive written final exam
Note that you must pass 13 of the 15 weekly lab exercises and also achieve a minimum score of 50% on the final exam to pass the course. Failure to do so will result in a failing grade.

Grading is on an absolute scale (no "curve"). The grade cutoffs are as follows based on the total weighted score for the course:

- A: 93 ... 100
- A-: 88 ... <93
- B+: 86 ... <88
- B: 80 ... <86
- B-: 78 ... <80
- C+: 76 ... <78
- C: 70 ... <76
- C-: 68 ... <70
- D+: 65 ... <68
- D: 60 ... <65
- F: < 60

For S/N grading, a satisfactory grade (S) requires a weighted score of 68 or above.

**Class Policies**

Please read and understand the following policies. To be consistent and fair with all students in the class, these policies will be strictly enforced without exception:

**Python and Computing Labs**

All labs and programming assignments are intended to be completed on CSE Labs UNIX machines; there are various CSE Labs throughout campus containing such machines available for your use. We will be using the Python 3 programming language for this class. Python is freely available for a variety of machines and you are welcome to do your individual work on the system of your choice, however your work will be graded using, and must function correctly on, CSE Labs UNIX machines. Therefore, if you work on your programming assignments using a different system, it is your responsibility to be sure that your program works on CSELabs machines before submitting it. Note that we do not provide any assistance for environments other than CSELabs machines.

**Grading Issues**

Grading is performed by class TAs and supervised by the graduate TA. If you have a question about grading, contact the graduate TA directly. It is your responsibility to report grading issues (missing or incorrect grades) within two weeks of the original posting date. Grade issues reported more than two weeks following the posting date will not be considered. Therefore, please promptly verify that your assignment and examination grades have been properly recorded on Moodle. It is your responsibility to provide proof of a missing/incorrect grade.

**Make-up Labs and Examinations**

Due to the physical limitations of the computer lab, make-up lab exercises cannot be provided regardless the legitimacy of absence. To accommodate situations such as illness, religious observance or University sponsored activities, up to 2 labs may be missed without penalty. If more than 2 labs will be missed, you have missed a significant portion of the course and will receive a failing grade.

Make-up exams will only be considered for legitimate absences defined by University policies. For a make-up exam to be considered, the student must contact the instructor at least one hour prior to the
start of the exam. Failure to notify the instructor prior to the start of the exam will result in an assigned grade of zero.

Late Policy for Homework Assignments
Completed Lab/Programming Assignments must be submitted on or before the assigned due date/time. Late assignment submissions will not be accepted. This means that there is no sliding grade penalty; late work will receive a score of zero. Programming assignments often take more time than anticipated, therefore you should plan to complete the assignment as early as possible to allow for contingencies such as illness, religious observance or other activities.

Incompletes
A grade of incomplete is generally not considered or granted. Incompletes will be given only in very rare instances when an unforeseeable event causes a student who has completed all the coursework to date to be unable to complete a small portion of the work (typically the final assignment or exam). Incompletes will not be awarded for foreseeable events including a heavy course load or poorer than-expected performance. Verifiable documentation must be provided for the incomplete to be granted, and arrangements for the incomplete should be made as soon as such an event is apparent.

Class Conduct
Students are expected to treat their fellow students in the class, the instructor, and the teaching assistants in a respectful manner. This includes arriving at class on time and staying until the end of class (arriving late and leaving early are distracting to your instructor and classmates, and they interfere with group work). Talking to neighbors, reading newspapers, using a laptop for anything but course-related work and sleeping during lectures are also distracting and disrespectful to others, so don’t do these things.

Withdraws
You are free to withdraw from the class up to the end of the eighth week of classes. Withdrawing thereafter is up to the college and is not automatic. If you are not doing as well as you had hoped in the course and are considering withdrawing, you should do so by the end of the eighth week.

Disability Accommodations
We desire to make learning rewarding and fun for all students and make every attempt to accommodate anyone who has a desire to learn. Students registered with Disability Services, who have a letter requesting accommodations, need to contact the instructor early in the semester to discuss the accommodations outlined in their letter (no later than 3 weeks prior to the first examination).

Disability Services (DS) is the campus office that works with students who have disabilities to provide and/or arrange reasonable accommodations. The DS website is: http://ds.umn.edu

Students who have, or think they may have, a disability (e.g. mental health, attentional, learning, vision, hearing, physical or systemic), are invited to contact DS to arrange a confidential discussion at 612-626-1333 (V/TTY) or ds@umn.edu

Student Mental Health and Stress Management
As a student you may experience a range of issues that can cause barriers to learning, such as strained relationships, increased anxiety, alcohol/drug problems, feeling down, difficulty concentrating, etc. These mental health concerns or stressful events may lead to diminished academic performance or reduce a student's ability to participate in daily activities. University of Minnesota services are available to assist you with addressing these and other concerns you may
be experiencing. You can learn more about the broad range of confidential mental health services available on campus via http://www.mentalhealth.umn.edu/.

**Scholastic Conduct:**
Academic dishonesty is a grave matter, and can result in failing the course and/or more severe disciplinary action. Academic integrity is essential to a positive teaching and learning environment. All students enrolled in University courses are expected to complete coursework responsibilities with fairness and honesty. Failure to do so by seeking unfair advantage over others or misrepresenting someone else's work as your own, can result in disciplinary action. The University Student Conduct Code defines scholastic dishonesty as follows:

- submission of false records of academic achievement
- cheating on assignments or examinations
- plagiarizing, altering, forging, or misusing a University academic record
- taking, acquiring, sharing or otherwise using course materials without faculty permission
- falsifying records or dishonestly obtaining grades, honors, awards, or professional endorsement, either acting alone or in cooperation with another
- use of a prohibited device during an examination

In the context of this course, we will specify whether assignments are individual or group work. For the rest of this section, “you” refers to you as an individual or you as a group, as appropriate, and “others” refers to other individuals or groups. The work you turn in for a graded assignment or exam must be your own. This means that on written assignments you must create and write your own work, and on programming projects you must design, implement, debug, and test your program on your own without the assistance of anyone other than the course instructor or TAs. In addition, copying, assisting, or collaborating on an exam is misconduct, as is changing your answer after the exam is returned and then asking for re-grading. Claiming emergency when none exists, in order to obtain special consideration on homework or exams, is also considered misconduct.

The following is a partial list of actions that are specifically related to the activities of this class that are deemed egregious misconduct:

**Plagiarism**
Webster's online dictionary defines plagiarism as: "a piece of writing that has been copied from someone else and presented as being your own work" or "the act of appropriating the ideas and language of another, and passing them for one's own".

Note that copying any portion of someone else's program for a homework solution, or submitting any portion of program code obtained from "outside" resources such as Internet web sites without including a proper citation is considered plagiarism and is a serious breach of ethical academic conduct. Doing so will result in sanctions ranging from a grade of zero for the assignment to failing the course.

**Unauthorized Disclosure/Distribution of Course Materials**
The University of Minnesota definition of Scholastic Dishonesty includes, "taking, acquiring, sharing or otherwise using course materials without faculty permission". Note that the materials provided in this course (including problem descriptions or portions of problem descriptions) are the property of the instructor and may not be shared with others or posted to Internet web sites without instructor authorization. Doing so is considered academic misconduct and will result in failing the course.
**Aiding Others in Academic Misconduct**
Aiding and abetting another student in an act of scholastic dishonesty is considered a serious offense. This includes showing or otherwise providing your program code or test solutions to another student. Note that you are required to electronically submit homework assignment solutions to be graded and checked using a software similarity measurement system. If the program source code you've submitted is subsequently determined to be a copy of another submission, *all* parties associated with the identified program codes will, at a minimum, receive a grade of zero for the entire homework assignment regardless who actually did the work. A second violation of this policy will result in an assigned grade of 'F' for the class. Simply stated, do not share your homework or exam solutions with anyone.

**Sharing Passwords**
In no case should your CSE Labs account and/or password be shared with anyone. Divulging your private account password to anyone else (including a lab partner) is a violation of the acceptable use policy for CSE Labs accounts and will result in account termination. In addition, knowingly divulging an account password to another person in order to share solutions to assignments, quizzes or examinations is a serious violation of acceptable scholastic conduct and will result in an assigned grade of 'F' for the class.

This is a partial list of specific examples and not intended to be complete. Any other egregious act of Scholastic Misconduct will result in failing the course.

It should be re-emphasized that **cheating, including two different people or groups sharing code for a programming assignment, will result in severe penalties, typically failing the course and a notification being sent to the CS Department and the CSE Dean.** If you have any questions about what is and is not allowable in this class, please ask the course instructor.