Syllabus	**Computer Science I: Programming Fundamentals**
CS 141	Spring 2010
CRN: 20274	

Time	Monday, Wednesday, Friday 9AM-9:50AM
Location	BARRET 035 (computer lab in basement of library)
Instructor	Eric Breck (`http://faculty.rhodes.edu/brecke`)
Office	419 Ohlendorf (901-843-3725)
Office Hours	Office Hours will be posted on my website; please send e-mail to make an appointment if you cannot make office hours. Tutoring will be available in Barret 033 Sunday, Tuesday, and Thursday evenings from 7-9PM.
Email	`brecke@rhodes.edu` To ensure a quick response, the subject line of your emails should read `cs141: [subject of question]` Do not assume I will read and reply to your messages instantaneously.

Moodle

All assignment submission will be via moodle. Do not e-mail me assignments. Course announcements and any updates to this syllabus will be posted via moodle.

Book

The course textbook is *Python Programming: An Introduction to Computer Science*, by John Zelle. Supplemental material may be distributed in class. Note that the book serves as a reference for the material we cover in class, but we will not cover material in the same order as it appears in the text.

Prerequisites

No previous experience in programming is required. You are expected to have a reasonable high-school mathematics background to appreciate the use of mathematical notation.

Course Description

CS 141 offers an introduction to the fundamental principles of programming, abstraction, and design. This course teaches you how to think as a computer scientist, by teaching the process of building abstractions to hide implementation details, and of controlling the intellectual complexity of designing large software systems by decomposing problems into simpler sub-problems.

Course Outline

The course will cover the following topics (not necessarily in this order):

- Program design
- Testing and debugging
- Variables, expressions and statements
- Conditionals and boolean logic
- Loops and iteration
- Simple graphics
- Functions
- Lists and strings
- File reading and writing

Python

This course will use the Python programming language as the vehicle for exploration of fundamental computer science concepts. However, this is not a course about Python; it is about the structure and interpretation of computer programs. Nevertheless, all programs assigned in this course must be written in Python.

The particular Python environment that will be used in this course is available in the computer labs on Rhodes College campus. Check the postings at Paul Barret Jr. Library for the hours of operation and locations of the on-campus computer labs.

You can download Python from `http://www.python.org/download/releases/2.6.2/`. You are free to develop the code for the assignments on your own computer using an environment of your choice. However, keep in mind that the source code that you submit for the homework assignments must execute successfully on the computers in the on-campus lab.

IMPORTANT NOTE ABOUT PYTHON VERSIONS. Python has very recently (December 2008) undergone a major version update, from version 2.6 to version 3.0. This is a non-backwards compatible change. Since version 3 is still relatively new, and since our textbook covers version 2, **we will only use version 2 in this course**. If you obtain your own version of Python for use on your own computer, make absolutely sure you are running version 2 (e.g. 2.5 or 2.6), and NOT version 3.0 or 3.1. Once you are comfortable with Python version 2, learning version 3 is not difficult, but it doesn't make sense to support different versions during this course.

Exams

There will be two midterms and one final exam.

- **Midterms**: Monday, 15 February and Wednesday 31 March, in class
- **Final Exam**: Monday, 3 May, 1PM
- Make-up exams will only be given in extreme circumstances.

Grade Breakdown

- 40% Programming Assignments (there will be 6-10 assignments).
- 30% Midterms
- 25% Final
- 5% Class Participation

Course policies

The instructor reserves the right to alter this syllabus as necessary.

Assignment submission

Assignments will generally be distributed on moodle immediately following the due date of the prior assignment.

The first line of each program source code file submitted for credit must be a comment that states your name and the assignment number. Assignments should be submitted via Moodle (by class time on the due date). Assignments received after class time on the date due are considered late. You are allowed to use the course textbook and the course notes for these programs. The use of any other material is forbidden.

Any assignments that require the submission of something other than program source code must be submitted as either plain ASCII text or Adobe PDF - no Word documents, please. To create PDFs on a Mac, choose Print from any application, then Save as PDF. To create PDFs on a PC, choose Print from any application, then choose PDFCreator as your printer of choice, and follow the prompts – this software is available on all campus computers, and can be downloaded free for your own machine. There are also scanners in the campus labs that can be used to create a PDF from a handwritten document. For **assignments that are not program source code only**, hardcopy submissions are also acceptable, though dispreferred.

Attendance

Attendance is expected at each class lecture period. You are responsible for learning any material presented during a class that you miss. Lecture notes and slides and textbook chapter references may be offered, but are not intended as a substitute for attending lecture – additional material may be presented beyond what is covered in these references.

Backups

Each student is responsible for keeping a back-up copy on disk of all files turned in for an assignment. Failure to do so could result in loss of credit for an assignment.

Classroom behavior

Please respect me and your classmates and do not browse the web, check e-mail, send or read text messages, or have private conversations during class.

Collaboration

You are expected to work individually on assignments. However, you are allowed and encouraged to discuss high-level details of the assignments. If group work is allowed, it will be mentioned explicitly in the assignment. Do not share your own code or copy others' code. Tools such as MOSS make it easy for me to check if you have copied code and just changed the whitespace and variable names; don't do it.

Communication with me

Like any human, my memory is imperfect. Therefore, if you make a verbal, in-person request or suggestion during class or elsewhere, you should follow up with an e-mail containing the same information. This includes but is not limited to: scheduling meetings, asking for regrades or other grade changes, indicating errors on the moodle site or in assignments, and anything else that requires me to remember to do something later.

Grade calculation

Grades for exams and assignments appearing in moodle are the recorded grades for each item. Aggregate grades appearing in moodle should be taken only as an estimate of the final grade. The final course grade will be calculated as specified above. Let me know as soon as possible if you believe a grade was recorded incorrectly or if you are unclear about how grades will be calculated.

Grading programs

Programs will generally be graded using the following guidelines.

- A (100 pts): The program is carefully designed, efficiently implemented, well documented, and produces clearly formatted, correct output.
- A- (94 pts): This is an 'A' program with one or two of the minor problems described for grade 'B'.
- B (88 pts): A 'B' program could easily have been an 'A' program, but it may have minor/careless problems such as poor, inadequate, or incomplete documentation; several literal values where symbolic constants would have been appropriate; wrong file names (these will be specified per program assignment); sloppy source code format; minor efficiency problems; etc. (This is not an exhaustive list.) You would be wise to consider 'B' the default grade for a working program — this might encourage you to review and polish your first working draft of an assignment to produce a more professional quality final version of your program.
- C (75 pts): A 'C' program has more serious problems: incorrect output for important special cases (the "empty" case, the "maxed-out" case, etc.), failure to carefully follow design and implementation requirements spelled out in the assignment description, very poor or inefficient design or implementation, near complete absence of documentation, etc.
- D (60 pts): A 'D' program compiles, links, and runs, but it produces clearly incorrect output for typical cases. Or, it may deviate greatly from the design or implementation requirements stated in the assignment description.
- F (35 pts): Typically, an 'F' program produces no correct output, or it may not even compile. It may "look like a program" when printed as a hard copy, but there remains much work to be done for it to be a correct, working program.

Late assignments

Late assignments will be accepted, with a penalty of 10% per day. Assignments submitted after a solution has been posted online or discussed in class will receive a 0. If a genuine emergency situation arises, please see me and we will work something out.

Network availability

You are required to take an exam at the specified time regardless of network accessibility. In other words, keep a copy of the practice tests, notes, etc. on **your** computer or network folder in case you have issues logging onto the network.

Numeric to letter grade conversion

Grades during the course will generally be given as numbers. The final course letter grade will be determined from the aggregate numerical grade using the following scale:

A	:	[93%, 100%]
A-	:	[90%, 93%)
B+	:	[87%, 90%)
B	:	[83%, 87%)
B-	:	[80%, 83%)
C+	:	[77%, 80%)
C	:	[73%, 77%)
C -	:	[70%, 73%)
D	:	[65%, 70%)
D-	:	[60%, 65%)
F	:	[0%, 60%)

Office Hours

Please come to office hours! You are not bothering me, I am here to help. During the office hours posted on my website, I will be in my office and available to answer your questions, help you with assignments, etc. You do not need to e-mail me to let me know you are coming. If for some reason I will be unavailable, I will try to send a notification via e-mail.

If you cannot make any posted office hours because of schedule conflicts, let me know by the end of the first week of class what your availability is, and I'll try to add another one.

You are also welcome to make an appointment, which consists of you sending me an e-mail saying "Can I come by at x time?", me responding "yes", and you showing up. Please do not miss such appointments.

Finally, you can also just stop by my office and see if my door is open. It often is, but won't always be.

Programming style

Programming is not a dry mechanical process but an art form. Well-written code has an aesthetic appeal while poor form can make other programmers and instructors cringe. Programming assignments will be graded based on correctness and style. Good programming practices usually include many of the following principles:

- Concise comments that summarize major sections of your code
  Comments should be correctly spelled and grammatical.
- Meaningful variable and function names
- Well organized code
- White space or comments to improve legibility
- Avoidance of large blocks of copy-and-pasted code

Scholastic Behavior and Academic Integrity

Plagiarism, cheating, and similar anti-intellectual behavior are serious violations of academic ethics and will be correspondingly penalized. If you are concerned about a possible violation of this kind, please talk with me. I understand the pressure that students may experience while at Rhodes, and I will try to help as best as I can.

All projects and tests must be the student's own work, unless otherwise instructed by your instructor. Copying all or part of a project assignment, or downloading code or text from the Internet and submitting it as your own, or having someone else write code or text for your assignment, or having someone else debug your assignment, or allowing someone else to copy from you, or writing or coding or debugging someone else's assignment — these are all included in the definition of reportable Honor Code violations for this course. If you have any doubts about whether or not a work practice on a project assignment is acceptable, please clear it with the instructor before proceeding.

Special Accommodation

If you are in need of special accommodations, please register with the Office of Student Disability Services (`http://www.rhodes.edu/disability`) as soon as possible so that all necessary arrangements can be made. Accommodations cannot be made unless you register; they also cannot be made retroactively.

Workload

It is important to stay current with the material. You should be prepared to devote regular weekly hours to this course. More specifically, you should devote at least 2-3 hours for each in class lecture. You should expect to spend significant amount of preparation for this course working on a computer to try out example programs and to develop the programming assignments. **Do not wait until the last minute to start your programming assignments.**