CS 1021C: CS 1 --- Spring 2012

Time: Lecture: MWF 1:25-2:20 PM, Lab: W 4:40-7:40 PM
Place: Lecture: RecCenter 3200, Lab: Old Chem 805

Professor: Paul Talaga PhD
Office: Rhodes 487
Office Hours: MT 2:30pm-4:30pm TR 9:00am-noon or by appointment
E-mail: paul.talaga@uc.edu
Website for course: Blackboard (blackboard.uc.edu)

Catalog Data:
CS1 - Introduction to computer science with emphasis upon structured and object-oriented programming, algorithm design, and problem-solving. Currently, Python is used as the computer language.

Prerequisites:
Pre-calculus mathematics

Textbook:
Guzdial & Ericson, Introduction to Computing and Programming in Python – A Multimedia Approach

Goals:
1. Students will know a standard core of syntax sufficient for programming in the course and using basic modern programming style: primary data types, representing integers, floating point numbers, boolean values, and character data; aggregate data types representing strings, vectors or arrays or lists, and collections (e.g., structures or records); control structures for choice/selection and iteration (or recursion); subprograms, including return values, parameter passing, recursion, and scope rules; and simple file I/O.
2. Students will demonstrate elementary understanding of the difference between editing, compiling, loading, and running programs.
3. Students will explain basic mathematical foundations of computing, to wit, limitations of integer and floating point arithmetic; writing a recursive function to implement a simple recurrence relation; and an elementary understanding of algorithm efficiency, e.g., binary search vs. linear search or computing powers.
4. Students, given problem statements, will design and code programs to solve the problems, including problems requiring nested control structures, designing several functions abstracting natural subtasks, functions calling other user-defined functions, and nested data aggregates.
5. The program designs will meet modern standards of design quality. Students will create them given only problem statements, not depending upon others to
suggest initial organizations, (e.g., useful functions) or to trace through execution on sample inputs. The programs will be correct.

**Topics:**

1. Intro to Computers and Programming.
2. Standard data types and their representations.
3. Selection control structures, including nested structures.
4. Repetition control structures, including nested structures.
5. Functions and parameter passing.
7. Recursion.
8. Arrays (or vectors or analogous structures) and Strings.

You are responsible for checking Blackboard for announcements, assignments, or any class changes.

**Laboratory Assignments:**

12 lab assignments will be given and should be completed during our weekly lab session. You need to hand in two (2) copies of your completed assignment. 1. Through Blackboard, and 2. Printed out and handed in at the end of our lab session. Please save paper and print multiple pages per sheet, duplex, and stapled. The printed version will be marked and returned.

Lab assignments are due on Blackboard and the paper version handed in by the end of lab. Late assignments will be deducted $2^{(d+3)}$ percent where $d$ is the number of days late from the due date/time. Time stops when files are deposited in Blackboard and hardcopy is submitted within 24hrs. As an example, if you submitted your lab on the night after lab, a $2^3$ (8%) deduction will be taken as long as you hand in your printed copy by the next night. Hand in late assignments in class or to my office until otherwise instructed. Only the last Blackboard submission will be graded.

The lowest lab assignment grade will be dropped.

**Homework Assignments:**

To augment the laboratory assignments, homework assignments may be given along with each lab. This assignment will be related to the lab, but longer and more in-depth. It will be due the following week, in class before lab. Just like the lab assignment, there will be a Blackboard submission and a paper-copy submission. The same late-policy applies to homework. The lowest homework grade will be dropped.

**Quizzes:**
Approximately 5 quizzes will be given, about every 2 weeks. The lowest quiz grade will be dropped. A quiz should take 30 minutes, about half a class period.

**Plagiarism:**

Students may discuss programming assignments with each other, but *may not* write *programs together or copy*. Any significant help, from student, book, web, etc. must be acknowledged in a comment in the program. Any duplication of code will result in a grade divided by the number of students copying. The Internet is considered a student in this case. Write your own code and you’ll be fine.

Cheating on a quiz or exam will result in an F for the course.

**Grading:**

- Class Participation: 5%
- Laboratory Assignments: 35%
- Homework: 20%
- Quizzes: 10%
- Midterm Exam: 10%
- Final Exam: 20%