Description

Prerequisite: Not a CS major or minor.
This course teaches hands-on computer programming skills to students who are not computer science majors. This includes how to frame a problem as an algorithm, how to translate an algorithm to executable code, how to be sure the code does what you want. Programming assignments are drawn from a variety of domains, and develops practical scripting and demonstration over theory. This course is not open to computer science majors or minors.

Objectives

The following are general goals for the course:

- Help you feel confident in writing computer programs
- Help you compete for jobs by providing computational skills
- Help you make effective use of computational methods in your chosen field

You will learn Python
The art of computational problem solving
Not memorizing facts or syntax

See What should I be able to know and do after this course? below.

Location

Class: MWF 11:00-11:50, Thompson 216

Text


Course Website

http://macs.citadel.edu/rudolphg/csci205/sp16

Grades will be posted in CitLearn.

Tools/Materials

- Free version of Anaconda Python tools. It includes a nice editor and package manager, among other things.
- CitLearn for submission of projects and assignments

Instructor Information

Instructor George Rudolph
E-mail george.rudolph@citadel.edu
Office 225 Thompson Hall
Phone 953-5032
Office Hours TBA; by appointment; when my door is open; email is a good way to reach me
Course Requirements

Read and prepare before class.
Ask questions in class. I will.
Work on code outside of class and in class.
Have fun with it. I will.

Some quiz questions and the assigned problem sets are borrowed or derived from materials available from the online course: John Guttag. 6.00SC Introduction to Computer Science and Programming, Spring 2011. (Massachusetts Institute of Technology: MIT OpenCourseWare), http://ocw.mit.edu (Accessed 31 Dec, 2015). License: Creative Commons BY-NC-SA

Quizzes and Homework (10%)

Open-book, open note pop-quizizzes may be given in class randomly. Homework and quiz questions will come from the readings and class discussions. Quizzes cannot be made up.

Problem Sets (70%)

The due date for each problem set is noted on the schedule.
Please note:
- ps0 is required but not graded
- ps1-3 and 10 are required
- choose any 3 of ps4-ps9 (denoted psX on the schedule) to complete, approximately 1 every 2 weeks

Some descriptions are long, but actual coding is not so long. Grok first the skill each problem set teaches—then solve the problem.

Late problem sets will be reduced by ½ a letter grade in points. Submissions more than 1 week late will receive a 0. If you submit a working, partial solution late, but within a week, the worst you can do is 50% of full points.

Final Project (20%)

The final project is your final exam for the course. It has two basic requirements: 1) solve a problem in a domain of your interest, and 2) use what you have learned in this class.
1. Submit weekly report each Monday in CitLearn discussion board—what you did on project that week
2. Prepare a 5-10 minute demo of your project
3. Submit any requested changes before final exam date Monday, May 2, 2015 at 4pm

Submission Policy and Process

Instructions for what to submit in CitLearn are included for each problem set. Please read the instructions carefully, as each problem set will have different instructions.
Overview

We will read Chapters 1-8, 10-15 from your text. Usually a quick read. The reading will be supplemented by hands-on work in and outside of class. If you need help, see your instructor, the Computer Science SI's, the Academic Support Center. Plan adequate time to succeed, and don’t stay stuck. There are a myriad of Python resources online—take advantage of them!

Tentative Schedule

<table>
<thead>
<tr>
<th>Sunday</th>
<th>Monday</th>
<th>Wednesday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 10</td>
<td></td>
<td>1: Intro.</td>
<td>2: Core Elements of a program; ps0 due</td>
</tr>
<tr>
<td>Jan 17</td>
<td>MLK Day: No class</td>
<td>3: Numerical Programs</td>
<td>3: Numerical Programs</td>
</tr>
<tr>
<td>Jan 24</td>
<td>Lab; ps1 due</td>
<td>4: Functions, Abstractions</td>
<td>4: Functions, Abstractions;</td>
</tr>
<tr>
<td>Jan 31</td>
<td>Lab; ps2 due</td>
<td>5: Types, Higher-order functions</td>
<td>5: Types, Higher-order functions</td>
</tr>
<tr>
<td>Feb 7</td>
<td>Lab</td>
<td>6: Testing, Debugging</td>
<td>6: Testing, Debugging;</td>
</tr>
<tr>
<td>Feb 14</td>
<td>Lab; ps3 due</td>
<td>7: Exceptions, Assertions</td>
<td>7: Exceptions, Assertions</td>
</tr>
<tr>
<td>Feb 21</td>
<td>Lab</td>
<td>8: Object-oriented Programming</td>
<td>8: Object-oriented Programming; ps3 due</td>
</tr>
<tr>
<td>Feb 28</td>
<td>Lab; psX due</td>
<td>10: Searching and Sorting (video lecture)</td>
<td>10: Searching and Sorting; (video lecture)</td>
</tr>
<tr>
<td>Mar 6</td>
<td>Lab</td>
<td>11: Plotting</td>
<td>11: Plotting</td>
</tr>
<tr>
<td>Mar 13</td>
<td>Lab; psX due</td>
<td>12: Stochastic Programs</td>
<td>12: Stochastic Programs</td>
</tr>
<tr>
<td>Mar 27</td>
<td>Spring Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apr 3</td>
<td>Lab; psX due</td>
<td>14: Monte Carlo Sim</td>
<td>14: Monte Carlo Sim</td>
</tr>
<tr>
<td>Apr 10</td>
<td>Lab</td>
<td>15: Understanding Experimental Data</td>
<td>15: Understanding Experimental Data</td>
</tr>
<tr>
<td>Apr 17</td>
<td>Lab; psX due</td>
<td>What do Computer Scientists do?</td>
<td>Project Demo</td>
</tr>
<tr>
<td>Apr 24</td>
<td>Project Demo; Class Ends</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May 1</td>
<td>Final Project Due, Monday, May 2, 4pm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GRADING

I do not “take off” points—you earn them. See article I Don’t Lie Awake at Night Thinking of Ways to Ruin Your Life.

QUIZZES AND HOMEWORK 10%
Unannounced quizzes given randomly during the semester. Homework assigned nearly once per week. Questions drawn from reading and class discussions. Cannot be made up.

PROBLEM SETS 70%
8 problem sets: ps0 (ungraded), ps1-3, 10 and any 3 of ps4-9 as scheduled.

FINAL PROJECT 20%
Choose a domain-project of interest. Work on it during the semester. Have fun, show off what you have learned. Due April 23, with final fixes due May 2.

TOTAL 100%

GRADING SCALE
A 90+ Above and beyond
B 80+ Beyond the norm
C 70+ The norm
D 60+ Not much there
F 59- Nothing done
Q & A Policies and Advice

What should I be able to know and do after this course (Student Learning Outcomes)?

- Write programs in Python to clean up or transform data.
- Write computer simulations and visualizations as an investigative tool
- Generalize the skills learned in this class to other tasks.

What is the course policy on technology use in class?

Follow the College’s policy on use of cellphones and electronic devices in class. In general, no electronic devices should be seen or heard in class, unless explicitly permitted by the instructor. You may bring a laptop to class, but use should be confined to note-taking, accessing course-related materials, or working examples. Any other uses are unauthorized.

What if I need disability accommodations?

If you have a documented disability, please see me at the beginning of the course so we can plan how to help you be successful.

What is the course policy on plagiarism, academic honesty and academic integrity?

Do your own work. You must demonstrate academic integrity when taking exams, doing assignments or programming. You must cite sources of information or ideas that are not your own. This includes clips or snippets of code from books or the Internet. You cannot turn in work someone else has done and claim it is your own work. Plagiarism is an automatic 0 for any quiz or assignment, may cause you to fail the course at the instructor’s discretion, and may incur further action by the college. When authorized to work in teams do your honest share of the work. Please know and follow the Honor Code and The Citadel’s Academic Integrity Policy. Ignorance is not an excuse.

What if I am feeling overwhelmed by this course?

Ask questions in class or in the lab. Most likely other students have the same questions. Come see the instructor during office hours or drop in—my door is open. Get help from the SI’s, the Academic Support Center or talk with classmates or upperclassmen who have been through the course.

What other resources might help me succeed in this course?

Talk to the Science Librarian in the Daniel Library. Visit the Academic Support Center. Make friends with a programmer.