

# CptS 481: Python Software Construction

## Spring, 2008

January 8, 2008

### Instructor

Bob Lewis  
Associate Professor, WSU Tri-Cities  
Office: West 134D  
(509) 372-7178  
[bobl@tricity.wsu.edu](mailto:bobl@tricity.wsu.edu)  
Office Hours: Tu 4-5pm, or by appointment

### Teaching Assistant

Lohith A. Rangappa  
Graduate Student, WSU Pullman  
Office: Sloan 334  
Lab: ETRL 301 (pervasive computing)  
Office Phone: ((509) 33)5-4160  
[lanusuya@eecs.wsu.edu](mailto:lanusuya@eecs.wsu.edu)  
Office Hours: M 11-12  
Lab Hours: W 11-12 (Sloan 353)

### System Administrator

#### Tri-Cities

David Search  
CIC 202

Submit system help requests through [tchelp@tricity.wsu.edu](mailto:tchelp@tricity.wsu.edu) or by phone: (509) 372-7334 (on campus: 2-7334)

#### Pullman

Submit system help requests through: <https://helpdesk.eecs.wsu.edu>, email [syshelp@eecs.wsu.edu](mailto:syshelp@eecs.wsu.edu) or by phone (509) 335-6773 (on campus: 5-6773).

### Lectures

MW 4:15-5:30pm, West 224 (Tri-Cities) and TERR 24 (Pullman)

In case of instructor absence or other unanticipated cancellations, make-up classes will be held at a time and place to be determined.

## Prerequisites

The course requires experience in an object-oriented programming language like C++ or Java, and some proficiency in computer use is expected. This should include, at a minimum, the ability to log on, send email, edit text files, and run programs. These requirements are satisfied by the lower division CptS sequence (CptS 121, 122, 223 and 224). As its course number suggests, even though Python is an excellent first programming language, the pace at which the material is presented makes this course inappropriate as a first programming course.

## Goals

This course will introduce the student to programming in the Python programming language and several major application packages. Particular emphasis will be made in using Python to design user interfaces with the Tkinter package. Quite apart from the language itself, the course focuses on how to use Python to integrate preexisting modules together to construct substantial applications that have a much larger scope of capabilities than the student has previously built.

## Text

Most readings will be in the official textbook

Mark Lutz, *Learning Python* (3rd. edition), O'Reilly

This text is in at the WSU bookstore.

Additional readings will be in these two references on Tkinter, freely downloadable:

Fredrik Lundh: <http://www.tricity.wsu.edu/~bobl/cpts481/an-introduction-to-tkinter.pdf>

New Mexico Tech: [http://www.tricity.wsu.edu/~bobl/cpts481/tkinter\\_nmt.pdf](http://www.tricity.wsu.edu/~bobl/cpts481/tkinter_nmt.pdf)

## Resources

### General Python Information

There is an enormous repository of Python information including documentation, source code, news, tutorials and examples at <http://www.python.org>. In particular, if Python is not already installed on your system or available from your operating system's own repository, you can download and install it (e.g. for Windows) from <http://www.python.org/download>.

### Course Library

Files from the books and miscellaneous examples are available under the course library maintained at <http://www.tricity.wsu.edu/~bobl/cpts481>.

### Course Syllabus

The syllabus is at <http://www.tricity.wsu.edu/~bobl/cpts481/syllabus.pdf> and will be continuously updated as the schedule evolves.

### Python IDEs

Although the instructor gets by with the traditional UNIX combination of a text editor (emacs) and the occasional shell (bash) window, you may prefer to develop your Python code with an integrated development environment. There are quite a few, and a good summary of them is at <http://wiki.python.org/moin/IntegratedDevelopmentEnvironments>. If you use an IDE, remember that you must still adhere to the rules given in the "Homework" section below.

# Outline

1. The Python Language (7 weeks)
  - (a) Orientation, Overview and History
  - (b) Types and Operations
  - (c) Statements and Syntax
  - (d) Functions
  - (e) Modules
  - (f) Classes
  - (g) Exceptions
  - (h) Built-ins and Common Libraries
2. Building User Interfaces with Tkinter (3 weeks)
  - (a) Widgets
  - (b) 2-D Graphics
  - (c) Screen Layout
  - (d) Events, Bindings, and Callbacks
3. Advanced Topics (3 weeks)
  - (a) Distributing Python Code
  - (b) Numerical Python
  - (c) Extending Python
  - (d) Case Studies
4. Selected Packages (2 weeks)  
possible topics (chosen with student input):
  - Databases
  - 3-D Graphics
  - Games
  - XML
  - Web Programming
  - Jython (interface to the Java runtime)
  - Multimedia
  - Parallel

## Schedule

This schedule will be revised as the course progresses. “—” indicates units continued from the previous session. Readings are from Lutz unless otherwise noted. Readings in brackets are optional.

Week	Mon	Topic	Wed	Topic
1	1/7	1a. Orientation, Overview, and History [ Chs. 1-3 ]	1/9	1b. Types and Operations Chs. 4-9 HW1 assigned
2	1/14	1c. Statements and Syntax Chs. 10-14	1/16	<b>HW1 due</b> HW2 assigned
3	1/21	<i>Martin Luther King, Jr. Day (no class)</i>	1/23	1d. Functions Chs. 15-17 <b>Quiz #1; HW2 due</b> HW3 assigned
4	1/28	—	1/30	1e. Modules Chs. 18-21 <b>Quiz #2; HW3 due</b> HW4 assigned
5	2/4	—	2/6	1f. Classes Chs. 22-26 <b>Quiz #3; HW4 due</b> HW5 assigned
6	2/11	—	2/13	1g. Exceptions Chs. 27-29 <b>HW5 due</b> HW6 assigned
7	2/18	<i>President’s Day (no class)</i>	2/20	1h. Built-Ins and Common Libraries <b>HW6 due</b>
8	2/25	2a. Tkinter Widgets Lundh & New Mexico Tech. <b>Quiz #4</b>	2/27	—
9	3/3	2b. 2-D Graphics HW7 (keyboard) assigned	3/5	2c. Screen Layout <b>Quiz #5</b>
10	3/10-14	<i>Spring Break (no class)</i>		
11	3/17	— <b>HW7 due</b>	3/19	2d. Events, Bindings, and Callbacks HW8 assigned
12	3/24	3a. Distributing Python	3/26	3b. Numerical Python <b>initial project proposal due</b>
13	3/31	3c. Extending Python	4/2	3d. Case Studies <b>HW8 due</b>
14	4/7	—	4/9	—
15	4/14	4a. Package TBD	4/16	4b. Package TBD
16	4/21	4c. Package TBD	4/23	4d. Package TBD
17	4/28-5/2	<i>Finals Week (no class, but <b>project due</b> at 5pm on day of final)</i>		

## Grading

The grade will break down as follows:

<i>Activity</i>	<i>Weight</i>
Homework	40%
Project	35%
Quizzes	25%

## Percentage-to-Letter Grade Mapping

This table maps an overall percentage  $p$  to the letter grade it will receive:

<i>Percentage Range</i>	<i>Letter Grade</i>
$90 \leq p \leq 100$	some kind of A
$80 \leq p < 90$	some kind of B
$70 \leq p < 80$	some kind of C
$60 \leq p < 70$	some kind of D
$p < 60$	F

As the course progresses, the instructor is free to revise these criteria downwards so that, for example, 87% might still result in an A-, but these ranges will not be raised.

## Homework

There will be several homework assignments requiring coding in Python. As the language is available for a wide range of platforms, students may work on their assignments on the platform of their choice. However, the grader will test all programming assignments with Python 2.5.1 under the Linux operating system, and students are responsible for making sure that their code works on that platform. Any extra steps the grader needs to make programs runnable under Linux will cause points to be deducted.

To reduce the chance of late assignments, homework must be submitted two ways:

1. As a **tarball** attachment to an email sent to the instructor prior to the start of class on the due date. After the first assignment, if you submit a tarball more than once per assignment, points will be deducted.

To test your program, the grader will detach the mail message, un-tar the attachment in an otherwise empty directory, and run the program with Python. Anything that requires more effort on the grader's part than this will cause points to be taken off. Special notes to the grader may be included in an optional README file, not in the body of the email.

2. Turn in a hard-copy printout of your code on the due date. (This is so the grader has something to mark up and return.) It is recommended that you use a "prettyprinter" such as **a2ps** to nicely typeset your Python code (do "**man a2ps**" on a Linux system to see how to run it).

Homework (email and hard copy) is due at the start of class on the given due date. Unless otherwise noted on the schedule. Late submissions are accepted up until the start of the second class period after the due date, with a 10% penalty.

## Project

Students will additionally complete a large-scale project in Python. An initial proposal must be submitted to the instructor via email no later than the date indicated on the above schedule. Work should not commence until the proposal has been approved by email from the instructor. A writeup (10-20 double-spaced pages, not including code) of the final project (with working demo emailed to the instructor in tarball form) will be due on the official day of the final. (Note: This class has no final.) No late projects will be allowed.

## Quizzes

There will be several short (30-min) quizzes to evaluate the student's reading knowledge of Python and the required reading. These will mainly be given during the coverage of Python and Tkinter prior to Spring Break. These take the place of a midterm.

## Policy on Academic Dishonesty

The fundamental requirement for all student work in this class is:

**Unless otherwise explicitly permitted by the instructor,  
all work you turn in must be your own.**

Any instance of academic dishonesty, as defined in the WSU Student Handbook, in this class will be dealt with severely (typically by failing the class) and reported to the WSU Office of Student Conduct. Note in particular that it is dishonest not only to copy another student's work, but to permit another student to copy yours.

Nevertheless, realizing that students can assist each other in understanding general course material, there are limited ways in which student collaboration is permitted:

1. You may communicate verbally with another student, as long as you do not communicate the answer or the other content of what you are going to turn in, whether code, mathematics, and/or text. A good way to work in this regard is for the student providing help to ask leading questions of the student needing help, letting them arrive at the answer themselves.
2. You may draw diagrams and such on a whiteboard, chalkboard, or piece of blank paper to illustrate the verbal points made in (1), as long as you do not write what you are going to turn in.

It will be up to the discretions of the grader (if applicable) and instructor to determine if any assignment shows evidence of collaboration beyond these limits. Any attempt to circumvent the spirit of these rules will be treated as a violation of the fundamental requirement. If you are in doubt, do not give help to or request it from another student: That's what office hours are for.

## Notice for Students with Disabilities

Reasonable accommodations are available for students who have a documented disability. Please notify the instructor during the first week of class of any accommodations needed for the course. Late notification may cause the requested accommodations to be unavailable. All accommodations must be approved each semester by (on the Tri-Cities campus) the Coordinator for Disability Services in room 269B West Building, 372-7351 or (on the Pullman campus) the Disability Resource Center, located in Administration Annex 205, (509) 335-1566. If you have not already met with the Coordinator this semester you will need to do so immediately.