CS-105L
Introduction to Computer Programming
using Python Spring 2015

Instructor:
Joel Castellanos
Office: Farris Engineering Center (FEC) 319
Office Hours: Wednesday: noon to 3:00 PM and by appointment.
e-mail: joel@unm.edu

Course Web site: http://www.cs.unm.edu/~joel/cs105

Course Description
CS-105L, Introduction to Computer Programming, is a gentle and fun introduction. Students will use a simple Integrated Development Environment to author small programs in a high level language that do interesting things.

Pre-Requisites: None.

Detailed Description:
CS-105L will be taught using the Python programming language. Python is a general purpose programming language that is relatively easy to learn. Python runs on Windows, Linux/Unix, and Mac OS. Python is free to use, even for commercial products, and is open source.

Students will author Python code using the educational Integrated Development Environment PyCharm. In addition to python programming, the course also covers some of the great ideas in computer science such as modeling, visualization, emergence, search engine page ranking systems, and complex adaptive systems. Throughout the course, students will author many short programs (usually less than fifty lines) that preform two-dimensional graphics, animations and customized image manipulations.

Much of the second half of the course will be devoted to group projects to be presented at Mesa del Sol Student Showcase to an audience of peers, UNM faculty from the School of Engineering, the College of Arts and Sciences, the College of Fine Arts, and Anderson School of Management together with people form the larger Albuquerque community. CS-105 is designed as a first course in computer programming for:

1. Pre-CS majors who do not have previous programming experience and are not yet ready for the fast pace and rigor of CS-152 (Computer Programming Fundamentals). Note: CS-105 does NOT count as CS credit for a CS major.
2. Students without programming experience who want to learn the basics of programming, an introduction to the Python programming language, and gain practical skills in using programs and scripts to create customized multi-media effects and other tasks.
Required Textbooks and Supplies


2. *Making Games with Python & Pygame* by Al Sweigart

3. The On-Line Python 3.2.6 Language Reference.
   Website: [https://docs.python.org/3.2/reference/index.html](https://docs.python.org/3.2/reference/index.html)

4. 2.0 USB (or newer) Flash Drive (1.0 is too slow) - Bring to every lab class.

5. i>clicker® (needed for lectures only, not labs). Available UNM bookstore.

   Note: this includes Python 3.4.
   Website: [https://www.jetbrains.com/pycharm/educational/](https://www.jetbrains.com/pycharm/educational/)

7. Pygame:
   For Windows Vista/7/8, install: pygame-1.9.1.win32-py3.2.msi
   Website: [http://pygame.org](http://pygame.org)
   For MacOS X, pygame does not exist for Python 3.x. However, there is a workaround. For detailed instructions, see the web:
   [http://florian-berger.de/en/articles/installing-pygame-for-python-3-on-os-x/](http://florian-berger.de/en/articles/installing-pygame-for-python-3-on-os-x/)

Grading

Each student's final course grade is a **weighted average** of four component grades:

- 50% Mini (lab) Projects.
- 10% Quizzes (in class i-clicker).
- 20% Exams (Midterm and Final).
- 20% Final Group Project

Note: When calculating your course grade, **DO NOT add mini project points to quiz points** or to exam points or to final project points!!! Points in these different categories are different units! For example, 5 quiz points may be worth 50 lab points (or maybe only 2 lab points).

The course letter grade is calculated from the numerical course grade by using the table below.

<table>
<thead>
<tr>
<th>Letter Grade Score Ranges</th>
<th>A+</th>
<th>A</th>
<th>90 - 92%</th>
<th>A-</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;100%</td>
<td>93 - 100%</td>
<td>B+</td>
<td>83 - 86%</td>
<td>B-</td>
</tr>
<tr>
<td>87 - 89%</td>
<td>77 - 79%</td>
<td>C</td>
<td>70 - 76%</td>
<td>C</td>
</tr>
<tr>
<td>77 - 79%</td>
<td>67 - 69%</td>
<td>D</td>
<td>60 - 66%</td>
<td>D</td>
</tr>
<tr>
<td>&lt; 60%</td>
<td>50 - 67%</td>
<td>F</td>
<td>0 - 59%</td>
<td>F</td>
</tr>
</tbody>
</table>
Lecture Attendance
Lecture class meets twice per week: Tuesday and Thursday 9:30 to 10:45. Lecture Attendance is a required component of the course. Quizzes, via i-clickers, will be given during almost every lecture. There are no make-up quizzes.

Lab Attendance
Lab class meets once per week in a computer lab. Lab attendance is taken both at the beginning and end of class. If you are absent, leave early or arrive more than ten minutes late, then you will be marked as absent. Each student may miss up to three lab classes during the semester without there being any direct effect on the grade. Each additional missed lab class will result in -2% to the student's final lab grade average. There are six lab sections at different times during the week. If for some reason you cannot attend your regularly scheduled lab class but are able to attend one of the other lab classes during the same week, then that other lab can count as your lab attendance.

NOTE: Before attending a different lab section, check with that section's lab instructor to make sure there is an open space for you.

NOTE: In order to receive credit for attending a different lab section, it is your responsibility to make sure the lab instructor of that section counts you as present while you are in the lab class (NOT after the fact). Your name will not be on that instructor's roster. You must make sure to speak to the lab instructor during the lab class, telling him or her first and last name, and in what section you are registered.

NOTE: The three lab classes that every student may miss without having final grade points deducted are designed to cover sports travel that prevents attending a different lab during the same week, short-term illnesses and other such events. A student that needs to miss many classes due to an extended or reoccurring illness or hospitalization will need to request a grade of Incomplete for the semester. With this, arrangements can be made for missed lab attendance and work to be completed during the following semester.

If you feel you need extra help or would simply like to attend lab section in addition to your own, then you are encouraged to do so. First, however, please contact the lab instructor of the extra lab you want to attend to make sure that there is enough space.

Late Policy
Lab assignments and projects can be turned in late with a penalty of FIVE PERCENT PER DAY (This would be -1 poin per day on a 20 point lab). Assignments more than 7 days late will not be accepted. This includes medically excused lateness!!! The primary reason for no negotiation in this is that solutions are generally released and discussed in class one week from the due date. There are, however, opportunities to
make up some missed work through extra credit assignments. A student that needs to miss many classes due to an extended or reoccurring illness or hospitalization will need to request a grade of *Incomplete* for the semester. With this, arrangements can be made for missed work to be completed during the following semester.

The lateness of an assignment is determined solely by the due date and the Blackbard Learn *timestamp of the final version* you submit.

When you submit an assignment in Blackboard Learn, it is ★YOUR RESPONSIBILITY★ to:

1. Exit Blackbard Learn,
2. Log back into Blackbard Learn,
3. Check that all required files are attached,
4. Check that the files uploaded correctly, and
5. Check that the contents of the submission are what you expect them to be. Do this by opening and examining your files from Blackbard Learn. Be sure to examine them carefully to make sure you submitted the correct version.

Up until the assignment due date, you can take back your submission, and resubmit. Doing this correctly is your responsibility and part of learning how to use computer systems.

**Academic Honesty**

Students are encouraged to help each other on labs through personal interaction and through the WebCT discussions. There is, however, a difference between helping and cheating. Cheating includes:

1. Copying another person’s work,
2. E-mailing or giving an electronic version of your work to anyone other than a course instructor.
3. Leaving a paper or an electronic version of your work where others can get it: you are responsible for your own computer security. **If you save a local copy of your work on a lab computer, delete it and empty the trash before logging off!**
4. Having another person complete any portion of your work.

The first time a student is caught cheating; the student will receive a negative grade for the assignment (i.e. if the assignment is worth 100 points, then a score of -100 is assigned).
<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>Installing Python, The Interactive Shell and Strings</td>
</tr>
<tr>
<td>Week 2</td>
<td>Guess the Number: import, random, loops, blocks, boolean data type, Condition Operators and Statements, <strong>while</strong> Statements, and Incrementing variables.</td>
</tr>
<tr>
<td>Week 3</td>
<td>Jokes: Making the most of <strong>print()</strong>, Escape Characters, Quotes and Double Quotes, the end keyword.</td>
</tr>
<tr>
<td>Week 4</td>
<td>Dragon Realm: Functions, <strong>and</strong> and <strong>or</strong> Operators, Truth Tables, Getting Player's Input, Return Values, Variable Scope, Parameters.</td>
</tr>
<tr>
<td>Week 5</td>
<td>Using the Debugger</td>
</tr>
<tr>
<td>Week 6</td>
<td>Flow Charts and Hangman: Lists, Concatenation, the <strong>in</strong> Operator, <strong>lower()</strong>, <strong>upper()</strong>, <strong>split()</strong>, <strong>range()</strong>, <strong>list()</strong>, for loops <strong>elif</strong>, <strong>random.choice()</strong>, and <strong>len()</strong></td>
</tr>
<tr>
<td>Week 7</td>
<td>Tic Tac Toe, Game AI, and Bagels</td>
</tr>
<tr>
<td>Week 8</td>
<td>Review and Midterm exam</td>
</tr>
<tr>
<td>Week 9</td>
<td>Cartesian Coordinates, Sonar and Caesar Cipher</td>
</tr>
<tr>
<td>Week 10</td>
<td>Graphics and Animation: Installing Pygame and Pygame Basics</td>
</tr>
<tr>
<td>Week 11</td>
<td>Memory Puzzle</td>
</tr>
<tr>
<td>Week 12</td>
<td>Slide Puzzle</td>
</tr>
<tr>
<td>Week 13</td>
<td>Simulate</td>
</tr>
<tr>
<td>Week 14</td>
<td>Wormy</td>
</tr>
<tr>
<td>Week 15</td>
<td>Squirrel Eat Squirrel</td>
</tr>
<tr>
<td>Week 16</td>
<td>Showcase final project at Mesa del Sol Written Final Exam</td>
</tr>
</tbody>
</table>