

# CS 111 – FOUNDATIONS OF PROGRAMMING

## SYLLABUS – FALL 2008

**Lecture:** MW 1:30 – 2:45 pm, MLH 310  
**Lab:** W 3:00 – 5:00 pm, MLH 310  
**Professor:** Dr. Holly Patterson-McNeill  
**Office:** TH 224  
**Phone:** 792-2342  
**E-mail:** [hapatterson-mcneill@lsc.edu](mailto:hapatterson-mcneill@lsc.edu)  
**Web Page:** <http://www.lsc.edu/hapmcneill>  
**Office Hours:** M-Th 9:00 – 10:00 am.  
**Also By Appointment**

### I. COURSE DESCRIPTION AND PREREQUISITES

CS 111 is an introduction to the fundamental concepts of computer science and programming, using a modern language and building toward an object-oriented perspective. Python is the language used in this course because it is easy to learn, has graphics capabilities, and is used in industry (as a scripting language, as a gaming language, as a multi-purpose language). You must have earned a grade of **C** or better in your core math course before beginning this course or obtain the permission of instructor. Yes, I am going to be checking. You must talk with me if you do not have the prerequisite math class.

**Credit Hours: 4**

### II. MATERIALS AND SUPPLIES

#### Textbook, Software, and Supplies:

- Zelle, John. 2004. *Python Programming: An Introduction to Computer Science*. Willsonville, OR: Franklin, Beedle & Associates.
- The software for this course is completely free.
  - Python 2.3.2 is available on the student disk supplied with your textbook.
  - Or, you can download Python 2.5.x from <http://python.org/>. Download the **Windows** version.
  - In addition, Python 2.5.x is installed in the library computer lab, on the school checkout laptops, among others.
- You need to save your programs to your own media. Your best bet is to use a flash memory device.

#### Resources:

- Python tutorial: <http://docs.python.org/tut/tut.html>
- John Zelle's 'Teaching with Python' page: <http://mcs.wartburg.edu/zelle/python/index.html>

### III. GOALS AND OBJECTIVES

#### Goals of the Instructor:

Your constructive feedback is always welcome to help me achieve the following goals:

- Provide all students the tools necessary to succeed in their pursuit of a high level of understanding of the principles of programming by:
  - Shifting ownership and responsibility for learning to the students
  - Using discovery, problem-based, and other active learning techniques
- To facilitate the improvement of student problem-solving skills and processes
- To facilitate an atmosphere conducive to learning the principles of programming for all students
- To provide quality feedback to students, enabling them to gauge their progress towards achieving their goal in learning the principles of programming

### Course Learning Objectives:

By the end of the course, the student should be able to:

- Implement Python programs with appropriate use of
  - functions
  - selection
  - repetition
  - data structures
- Apply object-oriented design methodology to solve a problem
- Work simple problems in base 2 and base 16 arithmetic
- Describe the basic design of a modern computing system

## IV. IMPORTANT NOTES

### Access & Accommodation Statement:

If you need course adaptations or accommodations because of a disability, if you have emergency medical information to share with me, or if you need special arrangements, please contact me immediately. I ask that you put any request in this regard in writing.

### Academic Dishonesty:

In the event of academic dishonesty, those involved **will receive an "F" grade for the course** and the violation of the Student Code of Conduct will be referred to the Director of Student Life for judicial action.

As defined in the [LCSC Student Handbook](#), Academic Dishonesty is:

- Cheating** - intentionally using or attempting to use unauthorized materials, information, or study aids in any academic exercise. The term "academic exercise" includes all forms of work submitted for credit hours.
- Fabrication** - intentional and/or unauthorized falsification or invention of any information or the source of any information in an academic exercise.
- Collusion** facilitating academic dishonesty - intentionally or knowingly helping or attempting to help another to commit an act of Academic Dishonesty.
- Plagiarism** - the deliberate adoption or reproduction of ideas or words or statement of another person as one's own without acknowledgment.

According to the Student Code of Conduct,

The sanctions imposed for a violation of this section of the Code are independent of, and in addition to, any adverse academic evaluation which results from the student's conduct. The course instructor is responsible for academic evaluation of a student's work and shall make that evaluation without regard to any disciplinary action which may or may not be taken against a student under the Student Code of Conduct.

**Incompletes:**

A grade of Incomplete can be granted only if 80% of the course work has been satisfactorily completed (passing) and if there is a documented family or medical emergency. In addition, you must communicate with me your desire for an incomplete and you must sign the Incomplete Grade request form that includes a mutually agreed upon date of completion and a list of work to be completed. An incomplete is NOT automatically granted. I do not grant a grade of incomplete if you fail to meet any of the above criteria.

**Office Hours:**

Office hours are done on a first-come first-served basis. BY APPOINTMENT is just that. You must make prior arrangements to meet with me. I read my **lsc.edu** e-mail several times daily; this is a very reliable means to contact me. My homepage contains a link to this class and its schedule of assignments. If you miss a class session, look at the class pages on the Web and BBCE to see what assignments were made.

**Respect for Others:**

Please demonstrate respect for the other students in this class:

- Turn off all audible pagers, cell phones and other such equipment while in the classroom.
- If you must take care of personal issues during class, please leave quietly and return quietly.
- If you must be late to class, please enter quietly, take your seat and get to work.
- If you know you must leave early, please let me know. Then leave quietly.

There is a pencil sharpener around the corner to the right of the stairwell. There is a uni-sex bathroom around the corner to the right of the stairwell. There is a ladies' bathroom to the right just before you enter this room.

**Computer Lab:**

The computer lab (MLH310) is open two nights a week starting after Labor Day.

**Important dates:**

<b>Aug 29</b>	Last Day to Register/Add/Drop Classes Online
<b>Sep 1</b>	Labor Day – <a href="#">no classes</a>
<b>Sep 2</b>	Instructor's Written Approval Required to Add Class
<b>Sep 8</b>	Last Day to Add Class(es) or to Drop without 'W' Grade
<b>Oct 1</b>	Last Day to Apply for Graduation Spring 2009
<b>Oct 24</b>	Midterm Grades Posted
<b>Oct 31</b>	Last Day to Withdraw from Class(es) or College for the Semester
<b>Nov 3-6</b>	Advance Registration for Spring 2009
<b>Nov 24-28</b>	Thanksgiving Break – <a href="#">no classes</a>
<b>Dec 15</b>	Final Exam

The last date to drop this class without a permanent entry on your transcript is **Sept. 8**. The last date to withdraw from this course with a W on your transcript is **Oct. 31**. After that date, withdrawal is permitted by petition only and approval of the division chair, Dr. Matt Johnston. The division chair requires documentation of **extraordinary** circumstances that prevented you from withdrawing by the deadline.

**V. METHODS OF INSTRUCTION**

I utilize a lecture and lab format. Programming and other work are assigned each week and sometimes each meeting. The lab meeting affords us the opportunity for hands-on work to test and apply material from

lecture. You will also be required to demonstrate mastery of the basic concepts in this class; that is, **you must pass all skill quizzes**.

You will not be 'lectured' about material that can easily be read from the book. The content provided in lecture will supplement the text material. You will have activities which require you to work through the material, learning to understand it in your own way.

### Tentative Weekly Schedule

Week	Topics
1	Introduction to Course, Python and Number Systems
2	Writing Simple Programs
3	Computing with Numbers and Expressions
4	Skill building with Numbers and Expressions
5	Computing with Strings
6	<b>EXAM ONE</b>
7	Using Objects and Graphics
8	Functions
9	Decision Structures
10	Loop Structures
11	<b>EXAM TWO</b>
12	Simulation and Design
13	<b>Thanksgiving Break</b>
14	Defining Classes
15	Data Collections
16	Object-Oriented Design
17	<b>FINAL EXAM</b>

A detailed schedule is available to you on BBCE.

## VI. PERFORMANCE CRITERIA FOR STUDENTS

- Demonstrate persistent and active learning through an organized approach that includes completion of activities, attending every class and lab period and bringing appropriate materials to support quality and active participation in lab activities, and appropriately using available resources to aid in learning course content.
- Develop high-level modeling skills where appropriate design elements can be quickly and correctly identified.
- Professionally perform computer skills and functions when applying computer science principles through correct use of equipment and software; and quality documentation of procedures such that others can easily read and assess the work performed.
- Exemplify effective communication skills as demonstrated through the use of appropriate technical language and as documented in assignments and problem solutions.

**Grading Breakdown for Activities Designed to Meet the Above Performance Criteria:**  
*Evaluation Methods:*

Programming	15%
Chapter Quizzes	15%
Skill Quizzes	10%
Two Exams (20% each)	40%
Final Exam	20%
<b>TOTAL</b>	<b>100%</b>

**Grading Scale** (expressed as percentages):

A	>= 94
A-	>= 90 to < 94
B+	>= 87 to < 90
B	>= 83 to < 87
B-	>= 80 to < 83
C+	>= 77 to < 80
C	>= 73 to < 77
C-	>= 70 to < 73
D+	>= 67 to < 70
D	>= 60 to < 67
F	< 60

## Description of Methods for Evaluation

**Programming (15%):** You will be assigned programs throughout the semester. Use the Program Formatting Information for each program as required before submission. The short, introductory programs will not require extensive documentation. I will clearly distinguish between the two types of programs. Even after successfully completing this course, you should not call yourself an expert programmer. That takes years of experience. You will be evaluated on 1) Completion, 2) Neatness and Organization, 3) Use of style, and 4) Correctness of your programs.. That means your evaluation is based on the number of programs you attempt, the level of completeness of each program submission, the neatness and organization of your work, the use of style in your programs, and how correct the programs are. Many of our programs will be graded during lab. Regular attendance is important. Under normal circumstances, if you have not completed your assignment by the due date, you should submit the work you have done for partial credit. For the sake of your grade, you should ALWAYS turn in SOMETHING. **Be sure to keep ALL graded material.**

Be sure to backup your work. USB devices fail; hard drives fail. Keep copies of your work on different devices. Failure to have your program ready because your drive crashed is not an acceptable excuse.

**Chapter Quizzes (15%):** This is a Web-enhanced course. After each chapter, you will have a quiz online using the BBCE system. These quizzes serve to guide your study for major exams. You will be given several opportunities (depending on the difficulty of the material) for you to improve your grade on these quizzes.

**Mastery Skill Quizzes (10%):**

There will be periodic, short skill quizzes during the semester to evaluate whether you have mastered the basic concepts of this class. During the last ten minutes of class, you will take a short skill quiz over a topic of the previous classes. To pass a skill quiz, your work must be perfect. If you make a mistake, you will retake the quiz at the end of a later class. Typically retakes are not given on the same day that you get the quiz back. Retakes may be completed at the end of class or during office hours or by special appointment, if necessary. If you are absent and miss a skill quiz, just complete the retake during regular class time. All skill quizzes must be completed by **5pm on December 12, 2008.**

**Semester Exams (20% each) and Final Exam (20%):**

We will have two semester (midterm) exams and the final exam. Because of the nature of programming, all exams are comprehensive. Each exam will emphasize certain chapters of our text for vocabulary and computer science concepts. But the programming portion is cumulative. What you have learned for programming in the earlier chapters of our text must be used in succeeding programs. Each semester exam will have an open book, lab exam in which you will have to write a program similar to what we will have practiced during the preceding weeks. This lab exam is not a separate exam, but is part of semester exam. The final exam is similar to the midterm exams. The final exam will be given during the scheduled final exam time. Do not make travel plans which require you to take the final exam early. No make-up exams will be given. If you have a serious conflict with an exam time, you must discuss it with me and take the exam early. Exams missed due to a serious illness or a family emergency (must be documented) will be dealt with on a case-by-case basis.

## VII. PROGRAM SUBMISSION GUIDELINE

A project report is a computer scientist's way of communicating the design and implementation of a problem solution to other computer scientists. For each major program done in the CS 111 course you will produce a program report. For the smaller programs, this will be part of the program documentation. All reports should include the following sections:

### **Cover Page with Problem Statement (10%)**

This consists of:

- Your name, the date, the course, and the assignment name or number in the upper half of the page
- A double-spaced, 12-point font problem statement in the lower half of the page.

The problem statement can be from one sentence to a single paragraph which describes the essence of the program. It should be written in your own words so that a fellow student, upon reading it, would understand the intent of the program. The problem statement must be clear and concise in describing what the program is about. There will be sample problem statements throughout the course.

### **Design Documentation (30%)**

This consists of:

- Analysis of the problem in the form of the Analysis Deliverable
- Output for at least two sample problems are worked
- Algorithm Design with associated design patterns
- Any appropriate diagrams

The design documentation will be very simple for the first few programs you write. As the programs get more complex, the design documentation will become more robust. The design documentation should be hard copy and not hand written, except for any diagrams.

### **Program Source (30%)**

This consists of:

- Source code with heading (programmers' name, date, purpose)
- Input files

The source code is expected to follow professional coding standards, be self-documenting, use an appropriate number of comments, and use whitespace to aid in readability.

### **Execution Results (30%)**

This consists of:

- Screen output
- Output files

Should your program produce merely error messages, you are expected to include only the first page of those error messages. If the output file is extensive, only the first page of that file should be included in this section.

**Note:** When requesting debug help from me, you must supply your design documentation and your algorithm walkthrough for or hand trace of the program. Examples of hand tracing will be included throughout the semester.

## VIII. APPENDIX

### Profile of Good Problem Solvers:

Computer scientists must be good problem solvers. Problem solvers:

- Have a strong desire and ability to identify and define the current problems confronting a process, system, person or group.
- Enjoy the problem solving process and as much as the solutions generated by the process.
- Seek to improve their skills by self-assessing their use of the problem solving process.
- Work to increase their ability to manage frustration so they can take on more challenging problems.
- Generate activities that produce opportunities to solve more intellectual problems.
- Adapt to and make effective use of changing technological environments.
- Seek out, learn, and use technological tools to improve the quality of their use of the problem solving process and the presentation of problem solutions.
- Logically identify key issues by utilizing previous problem solutions and/or outside expertise.
- Seek to improve the quality of problem solutions through the selection and use of better analytical modeling and tools.
- Partition a problem into a clear set of manageable subproblems.
- Produce multiple problem solutions
- Effectively integrate subsolutions into a cohesive solution.
- Assess the problem solution to make sure that every underlying assumption has been identified, tested, and documented.
- Make sure that every solution is tested for both reliability and robustness.
- Produce a documented problem solution that others can accept for quality.

---

26 August 2008