UPDATED Handout 1

# COMP 150: Introduction to Computing (Section 003)
# Spring 2009 Course Information & Syllabus

**Instructor:** R. I. Greenberg
Computer Science Department
Loyola University
Water TowerCampus, Lewis Towers 520
820 N. Michigan Ave.
Chicago, Illinois 60611-2147

**Phone:** (773)508-3782 on Monday/Wednesday/Friday
**Email:** rig@cs.luc.edu **Home page:** http://www.cs.luc.edu/~rig
**TA tutoring schedule:** See `http://www.cs.luc.edu/academics/services/tutoring`.

**Lectures:** MWF 10:25–11:15 pm in DH-342.
Sometimes lecture notes or a summary may be available on the web. Other than that, if you have to miss a class, get notes from another student; mine are typically pieced together from more than one place with a lot of metacomments, which makes it hard for anybody but me to follow them. Also get copies of any missed handouts (available on the web site). The handouts are numbered sequentially, starting with handout 0. On handout 0, you need to fill in some information and return it to me promptly so you can be on the email list and get access to the web site for the course.

**Office Hours:** In DH-225: 12:30–2:30 on Monday and Wednesday.
These are the guaranteed times to find me except as announced in advance. You should also feel free to look for me at other times or make appointments.

**Course Objectives:** This course provides a substantial introduction to computer programming as well as an introduction to a broad range of topics in the field of computing. The programming component is designed to provide a quick start by utilizing a simple but powerful language such as Python, and it will include an introduction to basic data types and operations, input/output, and flow of control. Broader topics in computing will include an introduction to digital logic and computer organization to explain the bridge between hardware and software, an introduction to historical and societal context, discussion of the power and limitations of computing, and possibly topics such as algorithm development and analysis, artificial intelligence, databases, and networking.

Outcome: Ability to write programs to manage and transform data; broad understanding of foundations of computing.

**Prerequisites:** None.

**Textbook:** Required:

Mark Guzdial. *Introduction to Computing and Programming in Python: A Multimedia Approach.* Pearson Prentice Hall, 2005.

Nell Dale and John Lewis. *Computer Science Illuminated.* Jones and Bartlett Publishers, third edition, 2007.

We'll also use parts of Dr. Harrington's Hands-on Python Tutorial at
`http://www.cs.luc.edu/anh/python/hands-on/index26.html` (available in webpage format and in printable format.)

Optional (comes bundled with Dale-Lewis from bookstore:
Bradley N. Miller and David L. Ranum. *Computer Science: The Python Programming Language.* Jones and Bartlett Publishers, 2007.

**Course Requirements:** There will be several homework assignments, three tests, and a final. The weightings within the semester grade will be: Homework 30%, Tests 1–3 15% each, and Final exam 25%.

**Homework:** Only homework turned in by the due date is guaranteed to be graded. Any special circumstances that cause difficulty in meeting the deadlines should be brought to the attention of the instructor in advance. Homework must be handed in at the beginning of class, since solutions may be discussed in the same class on occasion. Homework turned in to my mailbox will generally not be graded, since I do not check the box continually and cannot generally verify that homework was turned in before solutions were discussed in class. If you cannot turn in homework in person, you should put it under the door of my office.

**Exams:** The midterm exams, tentatively scheduled for week 6, week 8, and week 13, are 50 minutes long. The final exam is scheduled for 9:00 – 11:00 am on Monday, April 27.

**Collaboration:** *No* collaboration is permitted on exams. *Collaboration* on homework is acceptable, but *copying* is not! (Safeguard your files and printouts.) You may discuss solution techniques with other students, but you must write up your solutions independently. If you obtain a solution through research, e.g., in the library, credit your source and write up the solution in your own words.

**Tentative Course Outline and Approximate Schedule:**

The tentative schedule is given on a weekly basis.

1. (1/12) Administrivia. Syllabus and pair programming information at `http://www.cs.luc.edu/~rig/courses`. Overview of computing systems, history, and discipline of computing. CSI Chapter 1. Dr. Harrington's Hands-on Python Tutorial (for Python 2.6) through Section 1.2.
2. (1/21) Completion of CSI Chapter 1 and Python tutorial Section 1.2.
3. (1/26) Programming Introduction. Guzdial Chapter 2 and Python tutorial Sections 1.3–9. Data Representation, esp. Images & Graphics. CSI Sections 3.1 and 3.5.
4. (2/2) Modifying Pictures Using Loops. Guzdial Chapter 3.
5. (2/9) More on Loops. Guzdial Chapter 3 and Python tutorial Section 1.13. Binary Values and Number Systems. CSI Chapter 2.
6. (2/16) Exam I on CSI Chapter 1, Guzdial Chapters 2–3, and Python tutorial through Section 1.9. Modifying Pixels in a Range. Guzdial Chapter 4.
7. (2/23) Gates and Circuits. CSI Section 4.1–2 and first part of Section 4.4. Computing Components. CSI Chapter 5.
8. (3/9) Representing text, audio, and video. CSI Sections 3.3, 3.4, and 3.6. (The section on numeric representation goes deeper than we want here; you can do this and more in COMP 264.) Exam II on CSI Chapters 2 and 4, Guzdial Chapter 4, and Python tutorial Section 1.13.
9. (3/16) Modifying Sounds Using Loops. Guzdial Chapter 6.
10. (3/23) More on `if` Statements. Python tutorial Sections 3.1.1–5. `while` Statements. Python tutorial Sections 3.3.1–3. Creating and Modifying Text. Guzdial Chapter 10.
11. (3/30) Creating and Modifying Text continued. Guzdial Chapter 10. Making Text for the Web. Guzdial Chapter 11.
12. (4/6) Low-Level Programming Languages. CSI Chapter 7.
13. (4/15) Exam III on CSI Chapters 3 and 5, Guzdial Chapters 6 and 10, and Python tutorial 3.1.1–5 and 3.3.1–3. Low-Level Programming Languages continued. CSI Chapter 7.
14. (4/20) Artificial Intelligence. CSI Chap. 13.

**Core Learning Outcomes:**

This course has been approved for the QUANTITATIVE ANALYSIS Knowledge Area. (It also provides substantial support for the SCIENTIFIC LITERACY Knowledge Area, but has not been approved for that purpose by the core committee.) Approved Knowledge Area Learning Objectives are as follows:

QUANTITATIVE ANALYSIS Knowledge Area Learning Objectives:

- Represent and interpret quantitative information symbolically, graphically, numerically, verbally, and in written form.
- Recognize the limitations of mathematical and statistical models.
- Develop an understanding of the nature and history of mathematics, its role in scientific inquiry and technological progress, and its importance in dealing with issues in the public realm.

This course supports the following Skill Area Learning Objectives:

QUANTITATIVE AND QUALITATIVE ANALYSIS AND RESEARCH METHODS

- Represent and interpret quantitative information symbolically, graphically, numerically, verbally, and in written form.
- Recognize the power and limitations of mathematical and statistical models.
- Develop an understanding of the nature and history of mathematics, its role in scientific inquiry and technological progress, and its importance in dealing with issues in the public realm.

TECHNOLOGICAL LITERACY

- Demonstrate knowledge of the operation, application, and limitations of technologies important to his/her discipline.
- Select tools of technology appropriately in decision-making or to solve a problem.

INFORMATION LITERACY

- Demonstrate competence in using computer technologies (e.g., word processing, online discussion groups, software tools, library databases, and other research resources).

CRITICAL THINKING SKILLS AND DISPOSITIONS

- Analyze relationships among statements, questions, concepts, descriptions, or other forms of representation intended to express beliefs, judgments, experience, reasons, information, or opinions.