# COMP 125: Visual Information Processing (Section 001)
# Fall 2011 Course Information & Syllabus

**Instructor:** R. I. Greenberg
Computer Science Department
Loyola University
Water TowerCampus, Lewis Towers 524
820 N. Michigan Ave.
Chicago, Illinois 60611-2147

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

**Lectures:** MWF 10:25–11:15 am in CC-105.
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 LH-205: Mon. and Fri. 1:00–2:00 pm, Wed. 1:00–3:00..
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 an introduction to programming using App Inventor for programming mobile phones running the Android operating system. Topics to be covered include many standard programming language concepts such as variables, conditional execution, arrays, looping, and subroutines. At the same time, the course will work in the novel context of programming mobile phones and constructing programs by manipulating icons using a graphical user interface.

**Prerequisites:** None (but facility with basic high school mathematics is expected).

**Textbook:** David Wolber, Hal Abelson, Ellen Spertus, and Liz Looney. *App Inventor: Create Your Own Android Apps*. O'Reilly Media, 2011.

**Course Requirements:** There will be several homework assignments, a midterm exam, and a final project. The weightings within the semester grade will be: Homework 40%, "Midterm" exam 30%, and Final project 30%.

**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 or submit online.

**Exams:** The "midterm" exam is tentatively scheduled for week 11, is 50 minutes long and will actually be the only exam. Final projects will be presented during the final exam slot: 9:00–11:00 am on Monday, December 12.

**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:**

Recommended readings from the text are shown on a weekly basis. (When selected sections or subsections are listed, it is assumed that you will include the introduction of the corresponding chapter or section.)

1. (8/29) Administrivia. Building a first App. Through Chapter 1.
2. (9/7) Finish first App and look at "No Texting While Driving". Chapter 1. Skim Chapter 4, but especially look at using a TextBox for input.
3. (9/12) Paint Pot and Beginning Programming Concepts. Chapters 2. Background: Chapter 14, especially through "Event Types" section; Chapter 16, especially through "Setting and Getting a Variable" section.
4. (9/19) Paint Pot and Beginning Programming Concepts continued.
5. (9/26) Games. Chapters 3 and 5. Background: Chapters 17 and 18; also Chapter 23 on the orientation sensor and Chapter 21 on procedural abstraction.
6. (10/3) Games continued.
7. (10/12) Location sensing, etc. Chapter 7. Background: Chapter 23 first part.
8. (10/17) Using a simple database. Chapter 7 with Chapter 22 `TinyDB` portion. Software Engineering and Procedural Abstraction. Chapters 21 and 15 (debugging material also good to look at).
9. (10/24) Lists and Iteration. Chapters 8. Background: Chapter 19 before the section covering dynamic lists.
10. (10/31) Using Dynamic Lists, Recursion. Chapter 9. Background: Rest of Chapter 19. Exam review.
11. (11/7) Exam I on Monday. `foreach` Iteration and Using Web Databases. Chapter 10. Background: Chapter 20, rest of Chapter 22.
12. (11/14) More practice combining various control structures for Broadcast Hub in Chapter 11.
13. (11/21) Distribute and test loaner phones.
14. (11/28) Working with web information sources (Amazon API). Chapter 13.
15. (12/5) Catchup lectures or work on final projects.

**Core Learning Outcomes:**

This course contributes towards the following 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.
- May develop some understanding of the rudiments of statistics, including sampling and hypothesis testing, and the uses of statistical reasoning in everyday life.

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.
- May develop some understanding of the rudiments of statistics, including sampling and hypothesis testing, and the uses of statistical reasoning in everyday life.

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).
- Demonstrate knowledge of and ability to use a variety of methods, techniques, and databases in researching a topic.