

# EE 5453 (Sec. 2) --- Engineering Programming II (Fall 2014)

## Course Syllabus

**Instructor:**

Lide Duan  
Assistant Professor  
Department of Electrical and Computer Engineering  
University of Texas at San Antonio

**Office:** AET 2.376

**Phone:** 210-458-5208

**Email:** lide.duan@utsa.edu

**Office Hours:** 10:50 – 11:50AM MWF or by appointment

**Website:** All class materials will be handled electronically through Blackboard Learn

**Class Meeting Time and Location:**

4:00-4:50PM MWF, Business Building (BB) 3.03.14

**Course Description:**

Object-oriented programming for engineering design problems using Java and Python; scripting using Python; etc.

**Prerequisite:**

Graduate standing. Non-experience in Java or Python is assumed. However, it would be helpful if you have some basic knowledge in programming (in any language).

**Textbook:**

There's no required textbook. Lectures will approximately follow the references below.

**References:**

- Oracle Java tutorial: <http://docs.oracle.com/javase/tutorial/index.html>
- Oracle Java SE8 documentation: <http://docs.oracle.com/javase/8/>
  - Language Specification: <http://docs.oracle.com/javase/specs/jls/se8/html/index.html>
  - API Specification: <http://docs.oracle.com/javase/8/docs/api/index.html>
  - JDK Developer Guides: <http://docs.oracle.com/javase/8/docs/>
- Thinking in Java, 4th Edition: <http://mindview.net/Books/TIJ4>
  - Electronic book for purchase; source code for download.
- Python 3 documentation: <https://docs.python.org/3/>
  - Tutorial: <https://docs.python.org/3.4/tutorial/>
  - Standard Library: <https://docs.python.org/3/library/index.html>
  - Language Reference: <https://docs.python.org/3/reference/index.html>
- Dive Into Python 3 (free book): <http://www.diveintopython.net>

**Course Topics (tentative):**

- Java:
  - Introduction (history, editions, key features, object-oriented concepts, etc.)
  - Language Basics (variables, primitive data types, arrays, operators, expressions, statements, blocks, etc.)
  - Classes (objects, packages, access control, initialization and cleanup, etc.)
  - Inheritance (interfaces, reusing classes, polymorphism, nested classes, etc.)

- Collections and Generics
- Error Handling with Exceptions
- Java Standard Library (Numbers, Strings, java.lang, java.util, etc.)
- I/O, Concurrency, Graphical User Interfaces
- Python:
  - Introduction (history, key features and concepts, etc.)
  - Language Basics (statements, functions, etc.)
  - Data Structures (lists, tuples, sets, dictionaries, strings, etc.)
  - Modules and Packages
  - Classes and Iterators
  - Input and Output
  - Errors and Exceptions
  - Python Standard Library

### Grading Policy:

- Assignments: **30%**
  - 3 assignments (2 Java, 1 Python), each taking 10%.
- Exams: **35%**
  - Midterm (15%) + Final (20%)
- Project: **30%**
  - Proposal (5%) + Implementation (10%) + Final Report (10%) + Presentation (5%)
- Quizzes/Exercises: **5%**
  - The instructor will randomly choose a few classes to ask the students to turn in an answer to a very simple question. These are mainly for checking attendance.
- Total: **100%**

### About the Grading

- The final letter grades will be curved.
- After the grade of each assignment/quiz is posted, you will have a week to see me if there's any misgrading/miscalculation. After that, the grade is finalized.

### About the Assignments

- Each assignment will consist of several programming questions. Please straightly follow the naming/input/output requirements with each question, so that the graders can easily test your solution with unified test code (otherwise, no points will be given).
- Tentatively, the 3 assignments will be out in the 4<sup>th</sup>, 10<sup>th</sup>, and 14<sup>th</sup> week of the semester, and due in two weeks. No late turn-ins.
- All assignments must be completed individually. There's zero-tolerance on copying source code, either from online or from classmates. Both the copier and copiee will lose all the points for the assignment.

### About the Exams

- All exams are close-book and close-electronic device.
- No make-up exams (except for extremely special situation with legitimate proof and under discretion of the instructor)
- Tentatively, Midterm will be on Friday Oct. 10<sup>th</sup> during the class time.
- Final will be on Wednesday Dec. 17<sup>th</sup>, 3:15PM.

**About the Project:**

This is an individual project. Students will have large flexibility in choosing the project topic that they would like to work on. This can be an implementation of an existing paper scheme, a demonstration (with your own contribution) of an interesting feature, an extension to an existing framework, a development of a tiny app, and so on. The instructor will provide a few sample topics that the students can choose from. However, students are encouraged to propose their own topics, **as long as the topic is not directly covered in the class lectures**. The entire project should consist of the followings:

- **Proposal:** including the topic selection, literature search, references, work plan, etc. Tentatively, you will have about 4 weeks to finish the proposal.
- **Implementation:** Students are allowed to utilize an existing framework (if such one exists for the selected topic). If one chooses to do so, he/she must clearly identify what part of the project is implemented by him/herself.
- **Final Report:** How to run your project? What approaches were followed? What are the findings? Please complete it like a research paper, including Introduction, Mechanisms, Experiments, Results, Discussions, etc. You will have about 8 weeks after proposal to complete the implementation and final report (both are due after the Thanksgiving weekend).
- **Presentation:** Each student will have 5-10 minutes (depending on the enrollment) to present his/her work in the class. PowerPoint slides should be prepared. Tentatively, the last two weeks will be used for the presentations.