Fall 2015

# CSCI 2121

Zach Booth
*University of New Orleans*

# Software Design and Development II Lab

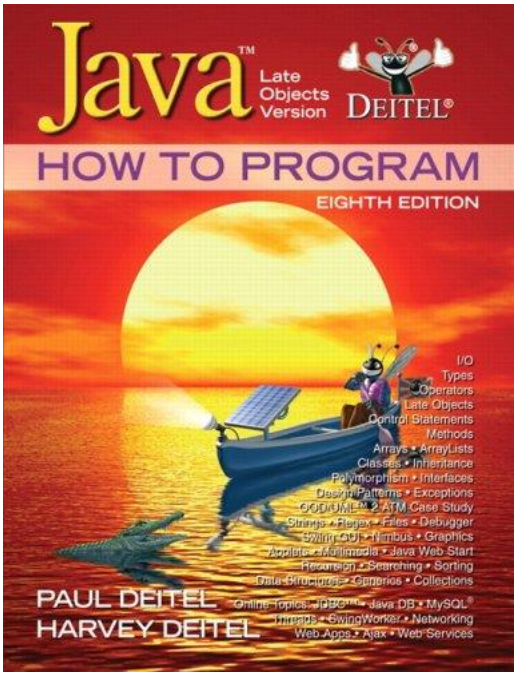
CSCI 2121 Section 001
Fall Semester 2015

Zach Booth
Office: MATH 321
email: zachb1988@gmail.com (preferred)

**Office Hours:** **M 8AM-11AM, T 3PM-6PM**; other times by appointment only. Office Hours will be held in **Math 321**.

**Prerequisite:** CSCI 1583 with a grade of C or better or consent of department; **concurrent registration in CSCI 2120 is required.**

**Text:** Dietel and Dietel, *Java, How to Program*, ***(Late Objects Version)*** *8th Ed.*




## Course Content:

This lab supports the introductory course CSCI 2120. We will be putting into practice the concepts covered in the lecture. The topics covered (roughly Chapters 12 through 25, omitting Chapter 16) will be (we reserve the right to adjust as the term progresses):

- Object-Oriented Design
- GUI Components
- Files, Streams, & Serialization
- Recursion
- Searching & Sorting
- Generics & Basic Data Structures
- Multithreading
- Networking

## Laboratory:

The purpose of the lab (CSCI 2121) is to give you an environment to try out concepts in software design via the development of software fragments with a lab assistant. Attendance and completion of lab work is **mandatory**.

## Grading:

(1) Laboratory work (CSCI 1581) will comprise 10% of your final grade. Each lab will consist of a set of typically between 5 and 8 exercises that you will have to submit through Git Lab. The exercises will be equally weighted on a 100 point grading scale. The labs will be graded for completion and correctness.

(2) You will receive the same grade for CSCI 1583 and CSCI 1581. Administrative constraints prevent us from offering the lecture and lab components as a single course. However, they are to be treated as such, hence the single, uniform grade.

(3) All work is graded on a numerical (percentage) basis. The correspondence between numerical and letter grades is given as follows:

A: >= 90,
B: 80 - 89,
C: 70 - 79,
D: 50 - 69,
F: < 50.

(4) It is expected that all homework will be turned in on time. Lateness penalties are:

- 1 day late - 10% off;
- 2 days late - 20% off;
- 3 days late - 40% off;
- >3 days late – not accepted

## Attendance:

The UNO Senate (Feb. 20, 2002) has made the taking of attendance a requirement for "developmental, 1000, and 2000 level courses." Attendance will therefore be taken at each class meeting. Although not a formal component of the computation of grades, good attendance will impact final grades in borderline cases. Important course content is often introduced outside of the published sources and/or scheduled presentations.

## Academic Dishonesty:

Finally, we must call your attention to the University's policies regarding academic dishonesty (http://www.uno.edu/studentaffairs/accountability.aspx). Academic dishonesty includes cheating, plagiarism, and collusion. In particular, it includes "the unauthorized collaboration with another person in preparing an academic exercise" and "submitting as one's own any academic exercise prepared totally or in part for/by another." In the event of academic dishonesty, **the student will be assigned a grade of 0** on the exam or exercise, the student will be informed in writing of the action taken, and **a copy of this letter will be sent to the Assistant Dean for Special Student Services**.

## Students with Disabilities:

It is University policy to provide, on a flexible and individualized basis, reasonable accommodations to students who have disabilities that may affect their ability to participate in course activities or to meet course requirements. Students with disabilities are encouraged to contact their instructors and/or the Office of Disability Services to discuss their individual needs for accommodations.

## Student Learning Outcomes:

At the conclusion of this course the students will be able to explain what Object-Oriented Programming is and will be able to implement complete computer programs using the object-oriented methodology. Students will be able to identify and explain the various control structures used in programming, will be able to explain what a method is and what happens when a method is called, and will be able to use arrays to process large quantities of data in programs. Students will also be able to apply software design techniques such as inheritance, polymorphism, and exception handling to produce extensible, easily maintainable, and robust programs.

## Tentative Schedule:

**WEEK 1 Aug 25-29**

Lab 1: Unit Testing with JUnit

**WEEK 2 Sep 1-5**

LABOR DAY

Lab 2: (Optional) Editing with Vim

**WEEK 3 Sep 8-12**

Lab 3: File Processing & Object Serialization

**WEEK 4 Sep 15-19**

Lab 4: Recursion

**WEEK 5 Sep 22-26**

Lab 5: Searching & Sorting

**WEEK 6 Sep 29-Oct 3**

Lab 6: Debugging with JDB

**WEEK 7 Oct 6-10**

Lab 7: The Generic Collections

**WEEK 8 Oct 13-17**

Lab 8: Writing Generic Classes
MID-SEMESTER BREAK

**WEEK 9 Oct 20-24**

Lab 9: Writing Basic Data Structures

**WEEK 10 Oct 27-31**

Lab 10: Git and Auto-documentation with Doxygen

**WEEK 11 Nov 3-7**

Lab 11: GUI's and the Model-View-Controller Pattern

**WEEK 12 Nov 10-14**

Lab 12: Advanced Git

**WEEK 13 Nov 17-21**

Lab 13: Multithreading

**WEEK 14 Nov 24-28**

Lab14: Network Connectivity
THANKSGIVING

**WEEK 15 Dec 1-5**

Lab 15: Database Connectivity

**WEEK 16 Dec 8-12 FINALS**