View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of New Orleans

University of New Orleans

ScholarWorks@UNO

University of New Orleans Syllabi

Fall 2015

CSCI 1583

Nathan Cooper
University of New Orleans

Follow this and additional works at: https://scholarworks.uno.edu/syllabi

This is an older syllabus and should not be used as a substitute for the syllabus for a current semester course.

Recommended Citation

Cooper, Nathan, "CSCI 1583" (2015). *University of New Orleans Syllabi.* Paper 162.
https://scholarworks.uno.edu/syllabi/162

This Syllabus is brought to you for free and open access by ScholarWorks@UNO. It has been accepted for inclusion in University of New Orleans Syllabi by an authorized administrator of ScholarWorks@UNO. For more information, please contact scholarworks@uno.edu.

Software Design and Development I

CSCI 1583 Section 601
Fall Semester 2015

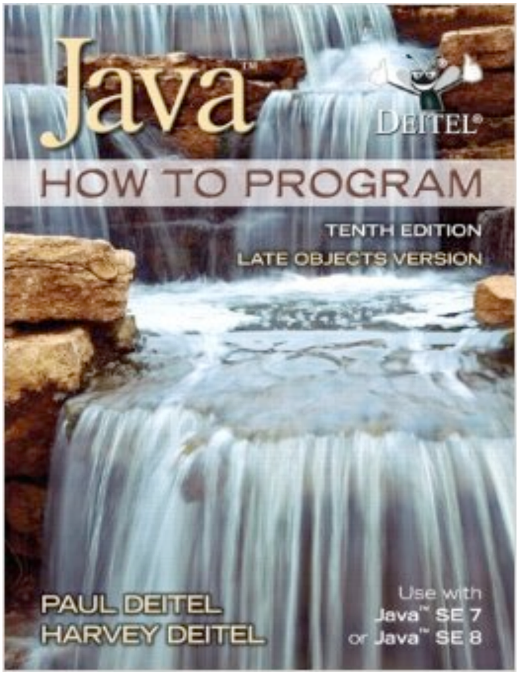
Nathan Cooper
Office: CERM 217
E-Mail: ncooper@uno.edu

Meets: **6:00 - 7:15 PM Tu/Th in Math 121**

Office Hours: **M/W 4:30 - 6:00 PM** or by appointment. Office Hours will be held in **CERM 217**.

Prerequisite: Math 1115 with a grade of C or better or consent of department; **concurrent registration in CSCI 1581 is required.**

Text: Dietel and Dietel, *Java, How to Program*, ***(Late Objects Version)*** *10th Ed.*



Course Content:

This course is an introductory course in Computer Science with an emphasis on programming in a high-level, object-oriented language. This supporting language is Java. The course is centered on the design and implementation of simple objects, and employs an iterative specify/design/implement/test strategy. The topics covered (roughly Chapters 1 through 11 and Chapters 14 and 15 from the text) will be (we reserve the right to adjust as the term progresses):

- Introduction to Computers and Software Development
- Control Structures and Algorithmic Thinking
- Methods and Structured Programming
- Arrays
- Strings and Files
- Classes and Object-Oriented Programming Concepts
- Inheritance
- Polymorphism
- Exception Handling
- Regular Expressions
- Object-Oriented Design

Laboratory:

The purpose of the lab (CSCI 1581) 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, homework/programming assignments **50%**, and tests **40%**. The test component will be computed as follows: Two in-class, closed-book announced tests plus the final exam grade **counted twice** gives four grades. The highest three will be used to compute the test component of your final grade. For example, if your in-class test grades are 70 and 80, and your final exam grade is 75, the grades 80, 75, 75 (highest three of 70, 80, 75, 75) will be used for the average. There will be weekly programming assignments.

(2) You will receive the same grade for CSCI 1583 and CSCI 1581 to be computed as described in the previous paragraph. 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

Note: We count school days (Sundays and holidays are not included).

(5) Homework Submission: Homework submissions through gitlab will be **required** for this course. Git is a tool commonly used by professional programmers for source code control, and you will be trained in the first week to install and use it. Effective use of our gitlab server will be the responsibility of the student. No exceptions. You will also be **required** to submit a hardcopy of your work. Failure to follow these requirements will result in a **grade of zero**.

(5) No make-ups for graded work (either tests or homework) will be given except for a legitimate (*e.g.*, medical) reasons.

(6) Questions about the grading of student work should be raised within 72 hours

of its return. After that time frame, issues raised will risk not being entertained.

(7) Students should retain all returned graded work, in case there are issues raised about the grade.

(8) The "I" grade (for Incomplete) is given only in exceptional circumstances, (*e.g.* missing the final exam because of a surgery).

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.

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.

Tentative Schedule of Study:

WEEK 1 (Aug 19-21) Chapters 1 & 2: Introduction
Lecture 1 Intro to Computers and Software, First Program, Hello World

WEEK 2 (Aug 24-28) Chapter 3: Control Structures Part I & Chapter 4: Control Structures Part II
Lecture 2 Memory Concepts, Arithmetic, Decision making with If statement
Lecture 3 Algorithms, Pseudo-Code, If, If…Else, While statements, Formulating Algorithms

WEEK 3 (Aug 31- Sep 4) Chapter 4: Control Structures Part II cont. & Chapter 5: Methods
Lecture 4 Compound Assignments, Primitive Types Counter, Controller Repetition
Lecture 5 Static Methods and Fields, Class Math, Declaring Methods, Method Call-Stack

WEEK 4 (Sep 7-11) Chapter 5: Methods cont.
Lecture 6 Argument Promotion, Random Numbers
Lecture 7 Scope of Variables, Method Overloading

WEEK 5 (Sep 14-18) Review for Exam & EXAM 1
Lecture 8 Review for Exam
Exam EXAM 1

WEEK 6 (Sep 21- 25) Chapter 6: Arrays, Strings, and Files
Lecture 9 Primitive vs. Reference Types, Arrays, Using Arrays, Examples
Lecture 10 Enhanced for, passing arrays to methods, Multidimensional Arrays

WEEK 7 (Sep 28- Oct 2) Chapter 6: Arrays, Strings, and Files cont.
Lecture 11 Variable Length Arguments, Command Line Arguments, Classes Arrays, ArrayList, Class String, String Methods,
Lecture 12 File Processing, Data Hierarchy, Sequential-Access Text Files, reading, writing files

WEEK 8 (Oct 5-9) Chapter 7: Intro to Classes and Objects
Lecture 13 Classes, Objects, Methods, Declaring Classes,
Lecture 14 Initializing Objects, Coding Examples

WEEK 9 (Oct 12-16) Chapter 8: Classes and Objects a Deeper Look
Lecture 15 Access Control, This, Time Case Study, Constructors
Break FALL BREAK HOLIDAY

WEEK 10 (Oct 19-23) Chapter 8: Classes and Objects a Deeper Look & Chapter 9: Inheritance

Lecture 16 Composition, Enumerations, Static, Final, and Package Access

Lecture 17 Inheritance: Superclasses and Subclasses, Protected, Relationships between Super and Subclasses

WEEK 11 (Oct 26-30) Chapter 9: Inheritance cont.

Lecture 18 Constructors in Subclasses, Software Engineering Issues

Lecture 19 Class Object, Coding Examples

WEEK 12 (Nov 2-6) EXAM 2

Lecture 20 Review for Exam

Exam EXAM 2

WEEK 13 (Nov 9-13) Chapter 10: Polymorphism

Lecture 21 Polymorphic Behavior and Inheritance, Interfaces

Lecture 22 Abstract Classes, Concrete Classes, final, Software Engineering and Interfaces

WEEK 14 (Nov 16-20) Chapter 11: Exceptions

Lecture 23 Error-Handling Overview, When to Use Exceptions

Lecture 24 Java Exception Hierarchy, finally block, Stack Unwinding

WEEK 15 (Nov 23-27) Chapter 11: Exceptions (Cont.)

Lecture 25 Declaring New Exceptions, Pre & Post Conditions, Assertions

Break THANKSGIVING HOLIDAY

WEEK 16 (Nov 30- Dec 4) Chapter 16 & Review

Lecture 26 Class String, Class Character, Tokenizing Strings, Regular Expressions, Classes Pattern and Matcher

Lecture 27 Review for Final

WEEK 17 (Dec 7-11) FINALS

Final Tuesday, December 8th, 8PM - 10PM