Intro to Software Design and Development with Java
Pilot Non-Credit Course
Spring Semester 2016

Aaron Maus
Office: MATH 312-C  Phone: (504) 357-9115
email: amaus@uno.edu (preferred)

Meets: Wednesday 6:30PM - 8:30PM in Math 320 (The Software Engineering Instructional Lab)
Office Hours: Monday 6:30PM - 7:45PM other times by appointment only. Office Hours will be held in Math 312-C.

Prerequisite: Ability to program using procedural programming methodologies. Should be proficient with variables, loops, selection statements, and methods.

Recommended 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 object-oriented systems and employs an iterative specify/design/implement/test strategy. The topics covered (roughly Chapters 7 through 12 and Chapters 20 and 21 from the text) will be (we reserve the right to adjust as the term progresses):
- Classes and Object-Oriented Programming Concepts
- Inheritance
- Polymorphism
- Exception Handling
- GUI Design
- Object-Oriented Design Patterns
- Generic Methods & Classes Generic Data Structures
- Generic Data Structures
**Grading:**

(1) Homework/programming assignments 60%, midterm exam 20%, and final exam 20%.

(2) 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.

(3) 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).

(4) 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. Failure to properly submit your assignments will result in grades 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.

**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 (Jan 11-15)** Chapter 7: Intro to Classes and Objects
Lecture 1  Classes, Objects, Methods, Declaring Classes, Initializing Objects

**WEEK 2 (Jan 25-29)** Chapter 8: Classes and Objects a Deeper Look
Lecture 2  Access Control, This, Constructors, Composition, Static, Final, and Package Access

**WEEK 3 (Feb 1-5)** Chapter 9: Inheritance
Lecture 3  Inheritance: Superclasses and Subclasses, Protected

**WEEK 4 (Feb 8-12)** Chapter 9: Inheritance cont.
Lecture 4  Inheritance hierarchies, Constructors in Subclasses

**WEEK 5 (Feb 15-19)** Chapter 10: Polymorphism
Lecture 5  Polymorphic Behavior and Inheritance, Interfaces

**WEEK 6 (Feb 22-26)** Chapter 10: Polymorphism cont.
Lecture 6  Abstract Classes, Concrete Classes, final, Software Engineering and Interfaces

**WEEK 7 (Feb 29-Mar 4)** Chapter 11: Exceptions
Lecture 7  Error-Handling Overview, When to Use Exceptions, Java Exception Hierarchy

**WEEK 8 (Mar 7-11)** Chapter 11: Exceptions Cont.
Lecture 8  Finally block, Stack Unwinding, Declaring New Exceptions, Pre & Post Conditions, Assertions

**WEEK 9 (Mar 14 - Mar 18)** Midterm
Lecture  Midterm
WEEK 10 (Mar 21-25) Break

WEEK 11 (Mar 28-Apr 1) Chapter 12: GUI Components
Lecture 9    Introduction to Java GUI Framework and Swing Components

WEEK 12 (Apr 4-8) Chapter 12: GUI Components
Lecture 10   Events, Observer/Observable, Model-View-Controller

WEEK 13 (Apr 11-15) Design Patterns
Lecture 11   Design Patterns with Examples: Strategy, Singleton, Adaptor

WEEK 14 (Apr 18-22) Chapter 20: Generic Classes and Methods
Lecture 12   Generic Methods, Overloading, Generic Classes

WEEK 15 (Apr 25-2) Chapter 21: Intro to Generic Data Structures
Lecture 13   Linked Lists, Stacks, Queues, Trees, Binary Trees

WEEK 16 (May 9-13) Final Exam
Lecture      Final Exam