

Object-Oriented Design and Programming I

CS 221-003 - Fall 2016

Instructor: George Thomas **Office:** Halsey 218
Email: thomasg@uwosh.edu **Phone:** 424-2069

Office Hours: MW 1:30-3:00,
TR 10:30-11:30,
Or by appointment

Section 003: Lecture: Tuesday 1:20-2:50 HS 237
 Lab: Thursday 1:20-2:50 HS 101C

Credits: This is a 3 credit course.

Prerequisites: A grade of C or better in Math 104 or Math 108 or Math 206 or Computer Science 142, or qualifying for Math 171 via the Mathematics Placement Exam.

Required Textbook: *Introduction to Programming with Java. A Problem Solving Approach*, John Dean and Raymond Dean, Second Edition, McGraw Hill.

Course Website: UWO D2L (<http://www.uwosh.edu/D2L>)

Note: If you have special needs, please do come and talk to me at the end of the first class.

Current Catalog Description

A first course in problem solving, software design, and computer programming using an object oriented language. Problem solving/software design techniques include: flow charts, pseudo code, structure charts, and UML class diagrams. Data structures and algorithms include: arrays, characters strings, linear search. Programming topics include; data types assignment statements, standard input/output, selection, repetition, functions/methods, parameters, scope of identifiers, debugging.

Course Outcomes

Students should be able to do the following upon successful completion of the course:

1. Given a description of a problem, apply the problem-solving steps used in computer programming to create a solution design.
2. Working from a solution design, implement a solution to a problem using the Java programming language.
3. Use incremental development to construct a working Java program.
4. Identify and apply appropriate data types within a Java solution.
5. Describe and identify key object-oriented programming concepts.

6. Differentiate between the memory allocation approach for primitive and reference data types in Java.
7. Examine the code available in the Java standard class libraries, and incorporate relevant Java standard classes into object-oriented design and program construction.
8. Create and document program design solutions for simple Java programs.
9. Given a solution design, create programmer-defined classes and incorporate these classes into Java program solutions.
10. Distinguish among the options for input and output using Java, and select appropriate approaches for a given Java solution.
11. Describe scope and persistence of objects and variables in object-oriented programming.
12. Identify and correctly apply sequence, selection, and iteration/repetition patterns in object-oriented Java solutions and program designs.
13. Identify and apply advanced class and object features, including: overloading methods and constructors, argument passing, object return from methods, and organizing classes into packages.
14. Manipulate collections of data using arrays and objects to solve a given problem using Java.
15. Describe the different sorting options available and select the best basic sort for use in a Java solution.
16. Apply test-first development to the construction of an object-oriented computer program.
17. Read and interpret UML 2.0 diagrams that document a problem, and implement the proposed solution using Java.
18. Implement professional standards and guidelines for designing and coding Java computer programs.
19. Present and justify, to a group of peers, the design and implementation of a problem solution.
20. Plan for and schedule adequate time to complete labs and projects no later than the required due date.
21. Consult various online and independent resources to independently attempt to resolve problems BEFORE requesting assistance from co-workers/co-learners or supervisor/instructor.
22. Determine when it is appropriate to seek assistance, from co-workers/co-learners or supervisor/instructor to resolve problems that could not be resolved independently.

Course Grading Policy

Your final grade for this course will be based on four components, namely exams, programming projects, homework, and quizzes and class participation. Your overall

numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

Component	Weight
Exams (3)	45%
Programming Projects	25%
Labs	20%
Quizzes and Class Participation	10%

Tentative Exam Dates are as follows:

- **Exam 1 – Thursday, 10/06 in lab**
- **Exam 2 – Thursday, 11/03 in lab**
- **Exam 3 – Thursday, 12/15 in lab**

Your letter grade for the course will be computed as follows:

Numerical Score	Grade	Numerical Score	Grade
>=92	A	72-78	C
90-92	A-	70-72	C-
88-90	B+	68-70	D+
82-88	B	62-68	D
80-82	B-	60-62	D-
78-80	C+	<60	F

While this overall grading scheme is fixed, I will be happy to discuss any issue you may have with individual grades. If you notice a mistake or have a question regarding a specific grade, please come and talk to me *as soon as possible*. Do not wait until the end of the semester to bring up grading issues.

Project and Lab Deadlines

Each lab will come with a deadline (day and time) by which it must be submitted. **Late lab submissions will NOT be accepted.**

Each project will also come with a deadline (day and time) by which it must be submitted. You are allotted *three* project credit days you can use through the semester. A credit day is exactly 24 hours or less. You can use unused credit days to submit a project after its deadline, without penalty. Any project submitted after the deadline, plus any credit days you have unused, will receive a zero.

For example, if you have 2 unused credit days available and a project is due on Tuesday at 5:00PM, you can submit it anytime by exactly Thursday at 5:00PM without penalty. Do note that if you submit your project on Thursday at 5:01PM, you will be penalized 100% of the score of the project and thus receive a zero! Note also that if you submit your project on Wednesday at 5:01PM, you will be charged two credit days (but no penalty, obviously).

Attendance and Participation

While attendance at lectures is optional, there will be frequent class participation exercises and unannounced quizzes to reinforce the material presented during lecture. You will, obviously, have to be present during lecture to get any credit for these. In summary, I do suggest that you not only attend **every** class meeting but also come **prepared** for and **participate** actively in it. I **strongly encourage you to ask any question** or raise any issue you have with the course either during or at the end of class, or during my office hours. I will also gladly meet with you by appointment.

Absences and Extensions

Extensions on deadlines may be granted at the discretion of the instructor if you provide a valid justification (in the form of a written excuse from a medical doctor or the Dean of Students Office) **before** the due date.

If you miss a scheduled exam (tentative dates are provided), you **may** be able to take a make-up exam provided you give the instructor a valid justification (see above) ahead of time if possible. Only one make-up exam will be given. It will be a comprehensive exam scheduled at the end of the semester. Similarly, there will be no make-up quizzes unless the instructor is provided with a valid justification (see above) for your absence on the day of the quiz.

Collaborating versus Cheating

Unless otherwise stated in the lab or project, all submissions must be entirely your own work. While it is acceptable to discuss the labs and projects at a high level (for example, at the design level) with others, you must submit your own work. **You may not “borrow” any piece of code or design of any length from someone else, the internet, or any other source, unless you can live with a zero and the other potential academic sanctions of cheating** (see [UWO Student Discipline Code 2007](#), Chapter UWS 14).

Tentative Calendar

Week	Date	Topics	Labs & Exams
1	9/07-9/09	Intro. to Computers and Programming (Ch.1)	Lab 1
2	9/12-9/16	Algorithms and Design , Java Basics (Ch. 2 and 3)	Lab 2
3	9/19-9/23	Java Basics - Operators (Ch.3)	Lab 3
4	9/26-9/30	Java Basics - Strings (Ch. 3)	Lab 4
5	10/03-10/07	Control Statements - Selection (Ch. 4)	Exam 1 (Thursday, 10/06)
6	10/10-10/14	Control Statements - Loops (Ch. 4)	Lab 5
7	10/17-10/21	Prebuilt Methods (Ch.5)	Lab 6
8	10/24-10/28	Object Oriented Programming (Ch.6)	Lab 7
9	10/31-11/04	More Object Oriented Programming (Ch.7)	Exam 2 (Thursday, 11/03)
10	11/07-11/11	More Object Oriented Programming (Ch.7)	Lab 8
11	11/14-11/18	Software Engineering (Ch.8)	Lab 9
12	11/21-11/22	Arrays (Ch. 9)	
	11/23-11/25	Thanksgiving Break	
13	11/28-12/02	Searching and Sorting (Ch. 9)	Lab 10
14	12/05-12/09	Multi-dimensional arrays and arrays of Objects (Ch.9)	Lab 11
15	12/12-12/16	Review	Exam 3 (Thursday, 12/15)