CompSci 221 Object Oriented Design & Programming I Fall'16

Instructor: George Georgiev
Office: HS, Room 217
Phone: 424 - 11 80 **E-mail:** georgiev@uwosh.edu

Office Hours: M W F: 9:10 – 11:20 or by appointment

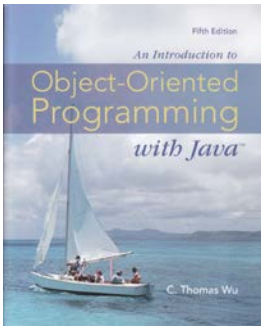
Section 1:

Lectures: MW 8:00 - 9:00 , NHS 237
Labs: F 8:00 – 9:00 , HS 101 C

Section 2:

Lectures: MW 11:.30 - 12:30 , NHS 237
Labs: F 11:30 – 12:30 , HS 101 C

Required Text: Wu C. T. (2006). *An Introduction to Object-Oriented Programming with Java*. McGraw-Hill. 5th edition.



References:
Lewis, J. and Loftus, W. (2009). Java *Software Solutions: Foundations of Program Design*. Pearson Addison Wesley. 6th edition.

Web site for the course: http://www.uwosh.edu/faculty_staff/georgiev/subjects/CSC221/

All materials: onD2L

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, 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.

Prerequisite: 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. (Fall, Spring) Required.

Class Number	61782	Career	Undergraduate
Session	Fourteen Week	Dates	9/7/2016 - 12/16/2016
Units	3 units	Grading	Student Option
Instruction Mode	In Person	Location	Oshkosh
Class Components	Lecture Required	Campus	Main

Course Outcomes

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.

Major Topics Covered in the Course

1. Introduction to Computational Thinking
2. Computer Programming as a Problem-solving and Design Process.
3. Introduction to Object-Oriented Programming Concepts
4. Objects, Classes, Members, Methods, and Standard Class Library
5. Abstraction and OO Program Design: Notation and Tools
6. Input/ Output in Java
7. Data Types and Persistence
8. Arithmetic Operators and Mathematical Functions
9. Creating and Using Classes in Java
10. Methods: Static, Parameters, Return Values, and Overloading
11. Sequence, Selection, and Decision-Making in Solutions
12. Iteration with Loops
13. One- and Two-Dimensional Arrays
14. Searching
15. Sorting

Course Requirements:

There will be three exams, unannounced quizzes, programming assignments, and laboratory works. The material for all exams will come from either a material covered in class, lab work, and/or programming assignments.

Complete all required work on time. In the event that an exam must be missed, or required work can't be completed on time, due to illness or other serious and unavoidable circumstance, notify the professor as far in advance as possible by phone or e-mail.

The programming assignments are due by the beginning of class on the due date (electronic copy (e-mail), and a paper (hard) copy of the assignment is due at the beginning of the class). Programs will be accepted up to three days late subject to the following penalties:

Turned in	Penalty
After 9:00 am on the due date	10%
1 day late	25%
2 days late	50%
3 days late	75%

Saturdays, Sundays, and holidays count when computing penalties.

If you work with a partner, you will submit one electronic copy and one paper copy of the assignment with all names on it. The partners will earn equal scores on the assignment. You may work alone on some assignments and with partners on others. You may change partners during the semester.

You are encouraged to discuss assigned problems with other people (teams) but you (team) must design and write own solutions/code for all assignments. Submitting modified versions of other people's (team's) work as own is considered cheating.

There will be no make up for unannounced quizzes.

There will be one make up for the exams, which will cover all topics. It will be at the end of the semester.

Make up exam will be given if you call before the exam, make arrangements, have a medical certificate signed by the physician, and have a note from the Dean of Students Office.

The three exams will be announced at least a week before taking place.

Laboratory assignments will be in the teaching lab. You are encouraged to discuss the lab assignment with others before and during the lab hours, but each student must demonstrate her or his own solution.

Evaluation:

Three Exams:		~60%	(20% each)
Programming assignments:	~25%		(equal points for each assignment)
Unannounced quizzes:		~5%	(equal points for each quiz)
Laboratory Assignments	~10%		(equal points for each lab)

Grading:

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

Feedback:

Your comments and questions about all aspects of the course (content, grading, teaching methods, pace, textbook, etc.) are welcome. You can use e-mail or talk to me during office hours.