Syllabus – CS 221 – Object-oriented Programming I – Fall 2017
MWF 9:10

INSTRUCTOR: Tom Naps
OFFICE: Halsey 214, phone 424-1388
EMAIL: naps@uwosh.edu
OFFICE HOURS: MWF 12:30 - 1:30, Thursday 10:00 - noon
COURSE DESCRIPTION: A first course in problem solving, software design, and computer programming using the Java language. Problem solving/software design techniques will be drawn from: flow charts, pseudo code, structure charts, and class diagrams. Data structures and algorithms include: Arrays, character strings, searching, and sorting. Programming topics include: data types, assignment statements, standard input/output, selection, repetition, functions/methods, parameters, scope of identifiers, debugging.
REQUIRED ONLINE TEXTBOOK: We will be using a highly interactive online book (a "Zybook") for the course that is available from Zyante Publishing. In addition to providing high-quality textual descriptions, Zybooks require you to be an active participant in your learning by engaging in many interactive learning activities. Your book should be ready for you by August 21, 2017. Between then and the beginning of the semester subscribe to your book by:

1. Signing in or creating an account at `learn.zybooks.com`
2. Entering zyBook code UWOSHCOMPSCI221NapsFall2017
3. Clicking Subscribe

The cost to subscribe is $67; any applicable returning student discounts will be applied automatically. Your student subscriptions will be valid through 01/04/18. You may also print PDF versions of each chapter of the book to study from when you are offline. Although the PDF version will not have all the interactivity of your online subscription, it provides a form of permanent access to the book's base material. Should you have any questions regarding the book, you should find a "Need help" selection in the pull-down menu for your profile in the top right corner of the book's login page (see attachment). Clicking that will get you in touch with the support team for the book, and I have found them to be very responsive to questions you might send them.

OTHER REFERENCES: Daily class handouts available by 8:00pm on the day preceding each class on D2L.

Learning Outcomes – You will be expected to . . .

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 Policies

Your grade for the course will be based on the following weighted factors:

Factor:	**Weight:**
Performance on assigned *Participation Activities* in your Zybook	5%
Performance on assigned *Challenge Activities* in your Zybook	5%
Performance on assigned *ZyLab Programming Problems* in your Zybook	5%
Performance on *POGIL (Process Oriented Guided Inquiry Learning) Activities* that we do in class	5%
Performance on lab exercises that you work on during our Monday lab period	10%
5 Programming Projects	25% in total
Three Exams:	45% in total

To be sure we understand what is involved in each of these factors ...

- A Zybook participation activity (ZPA) is an activity used in initially learning a topic. A ZPA is a more engaging way of doing your reading assignment before our class period begins. You can get all ZPA points just by completing the assigned activities. If a ZPA involves answering a question, there is no penalty if your answers are wrong the first time (or multiple times) or if you show yourself the answer. Because ZPAs are essentially "reading on steroids", they will be assigned and must be completed *before* we cover that material in greater depth in class. The ZPAs that must be completed before a given class will always be posted in the D2L calendar for CS 221.
- A Zybook challenge activity (ZCA) is a homework problem that is assigned for practice with a concept *after* we have finished covering it in class. Assigned after a class, they must be completed before the next class meeting for you to get credit. Many of the ZCAs are "progressive" in that they require you to work through several levels before you receive full-credit. Others will require that you write a code segment to complete the solution of a particular problem. The ZCAs assigned after a class and due before the next class meeting will always be posted in the D2L calendar for CS 221.
- A Zylab programming problem (ZPP) will require you to write a relatively small complete program to solve a particular problem. As we progress through the term, you will find that I have spliced them in at appropriate sections of the book. To receive any credit, they must be completed by the due date specified in their Zybook description. The due dates for a ZPP will also be posted in the D2L calendar for CS 221.
- A POGIL activity will often occur during our Wednesday or Friday class period when you will be assigned to a small group to work collectively on solving some problems. You receive full credit for the activity by being a willing and productive member of your group. If you miss the class when a POGIL activity occurs, you receive no credit for it.
- Each Monday, we will meet in the Halsey 101 Teaching Lab, where you will work on solving several lab exercises under the guidance of myself and Julia O'Connell, the course lab assistant. By preparing well for the lab, you should be able to finish all the exercises before the end of the lab period. If you don't, you may still receive credit for them provided they are completed before the beginning of the lab period on the following Monday.
- Programming projects require that you develop a substantial program, tying together many concepts, to solve a more difficult problem than what you encounter in lab exercises or ZPPs. Each project will have a deadline that allows you about 10 days to work on it. If you get started in a timely fashion, you should be able to complete it. However, if you procrastinate working on the project, disastrous results are likely. Late programming projects will be accepted but will be penalized at the rate of 10% of their point value the first day late, *an additional* 20% the second, *an additional* 30% the third. Any programming project submitted more than three days after the deadline will not be evaluated and will receive zero credit.

  Your work on programming projects is to be done without consultation or help from other students. 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 – `http://www.uwosh.edu/dean/conduct.htm`.
- Each exam will involve a mix of multiple-choice and written problems for which you should be thoroughly prepared by your timely and conscientious efforts on ZPAs, ZCAs, ZPPs, POGIL activities, lab exercises, and programming projects.

  If you are unable to take a scheduled exam, it may be possible to take a make-up exam provided that you do BOTH of the following, which are then subject to my approval:
  - Make arrangements prior to the scheduled exam (for last minute emergencies, telephone me at 424-1388 or leave a message at the computer science office, 424-2068). No after-the-fact notifications will be accepted ... *AND*
  - Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

  Only one make-up exam will be given. It will be a rigorous comprehensive exam given at an arranged time during the last week of the semester.

  To get the 45% contribution to your grade from the three exams, I will use the formula:

$$E = \frac{2}{9} \times E_{worst} + \frac{4}{9} \times E_{best} + \frac{1}{3} \times E_{other}$$

  where E_{worst} is your worst exam score and E_{best} is your best exam score. Essentially this lessens the effect of a bad exam score by only counting that exam half of your best exam score.

At the end of the term, your work in all of these areas will contribute to a numerical grade for the course based on a 100-point scale. Grade cutoff levels on this final scale are:

A ≥ 92	B ≥ 82	C ≥ 72	D ≥ 62
A- ≥ 90	B- ≥ 80	C- ≥ 70	D- ≥ 60
B+ ≥ 88	C+ ≥ 78	D+ ≥ 68	F < 60