CS 221: Object-Oriented Design & Programming I

Spring 2021

Instructor: Michael P. Rogers
Office: Halsey 214
Office Hours: TuTh 3:00-5:00 PM, Fr 1:30-2:30 PM (via Collaborate Ultra & Zoom)
Email: rogersm@uwosh.edu
Phone: 920-424-1388 / 309-825-6454

Class Times: MWF 11:30 AM-12:30 PM, Halsey Lab 101C
Credits: 3

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.

Delivery: This class will be taught using the HyFlex model - you may participate face-to-face or online, and transition between the two at your discretion.

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 Website: if it happens in this course, it will be posted on UWO's Canvas site. Set up notifications to be alerted when announcements are posted, new assignments, quizzes, or notes are posted and graded, etc.

Required Textbook:
COMP SCI 221: Object-Oriented Design and Programming I, Online book by Zybooks.

Subscription Instructions:
1. Sign in or create an account at learn.zybooks.com, using your uwosh.edu email address
2. Enter zyBook code: UWOSHCOMPSCI221RogersSpring2021
3. Subscribe ($58, you can start subscribing on Jan 19, 2021)

Course Outcomes :
Upon successful completion of the course, students will be able 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.

Grading Criteria:

Category	%
Exams (3)	45
Labs	15
ZyBook - Participation & Challenge Activities	10
Projects	25
Quizzes	5

Grade Scale:

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

Exam Dates (tentative, subject to change):
Exam 1 - March 10
Exam 2 - April 21
Exam 3 - May 14

Late Policies: Douglas Adams once said, "I love deadlines: I love the whooshing sound they make when they fly by". Hopefully you'll agree with the first part, and not the second, but regardless the policies are as follows:

1. Exams - If you are unable to make an exam, a) let me know in advance, via email, and b) provide justification (a note from medical professional who treated you, or a representative from the Dean of Students Office).
2. Labs, ZyBook Exercises - late labs and ZyBooks exercises will not be accepted (plan accordingly!)
3. Projects - you will be provided with 3 virtual "Whoosh" cards at the beginning of the semester. Each can be redeemed in exchange for an extra day for a project. Do not squander them - late projects will not be accepted without a Whoosh card.

Absences: It has been scientifically proven that the most significant factor for predicting student success is attendance (although whether this is truly causation or merely correlation is another question). Students should attend each and every class, and notify the instructor ahead of time if you will be absent. Attendance may be taken and factored into the Participation category.

Academic Integrity: In the same way that you can't get in shape by having your friend run round the block, or experience bliss by watching somebody else complete a crossword puzzle, you cannot learn by copy-and-pasting somebody else's answers.

First, try to solve any problem that you are presented with entirely on your own, without anything other than your text/notes. If you can do that, you truly understand the material. If not (and this definitely will happen unless you're some sort of prodigy), ask your instructor: they wrote the question, they probably know the answer :-).

You may have a *general* conversation (in English, pseudocode, or by drawing pictures) about how to approach a problem, and you may have a peer examine your code to help debug it. You may not look at or use your peer's code, nor turn in code that you find elsewhere: such plagiarism will result in a 0 for the assignment, and other unpleasantness may ensue. See the Dean of Students Office web page for more information.

Exams and quizzes are to be done entirely on your own. These are closed book, closed notes, because this material is so absolutely fundamental that you must have it at your fingertips.

Accessibility: Your instructor is committed to ensuring a fair playing field. If you have a disability and need assistance (e.g., a note taker, certain seating, extra time to take tests, adaptive technology, etc.), please register with the Accessibility Center, and we work hard to accommodate your needs.

Non-discrimination and Anti-harassment: Your instructor is committed to maintaining a harassment-free, welcoming classroom, and will not tolerate discrimination on the basis of race, religion, creed, color, sex, gender, identity/expression, ancestry, national origin, age, marital status, relationship to other employees, sexual orientation, disability, veteran's status,

membership in the military, arrest/conviction record, political affiliation, or any other protected status.

Feedback: Your instructor thrives on feedback. If a concept doesn't make sense, **ask**, and if the answer didn't make sense, **ask again**! There are no stupid questions (and any stupid answers are my responsibility 😀). It may seem intimidating to put yourself out there and admit that you don't understand something, but:

1. if you didn't understand something, there is a good chance that your peers didn't either, and they will (silently, or better yet, by banging on desks like they do in parliament) applaud your act of inquiry
2. you needn't worry about "slowing down the class". Learning never slows down anything.
3. your instructor is intimately familiar with being in situations where things don't make sense: you will receive a very sympathetic hearing

If you do feel uncomfortable asking in class, please ask during office hours.

COVID-19 Policies:

1. Everybody must
    1. wear a face mask (covering both mouth and nose), inside all University buildings;
    2. practice social distancing, and
    3. practice appropriate hygiene, i.e., wash your hands (see, you mother *was* right :-)
2. This is a HyFlex class: you may attend face-to-face, or online, and transition between the two as you see fit.
3. The class will be held synchronously: students attending online will connect via Collaborate Ultra during class time.
4. Online students will be expected to take exams during the same time, and in the same fashion, as the in class students.
5. Students who are feeling poorly are asked to follow the advice listed on the Titans Return page.