

Introduction to Computer Science and Programming

Computer Science 142

Instructor:	Erik Krohn
E-mail:	krohne@uwosh.edu
Text Message Only:	608-492-1106
Class Times:	Tuesday: 5:10pm - 7:10pm
Classroom:	Halsey 208
Lab Times:	Tuesday: 6:10pm - 8:10pm
Lab Room:	Halsey 101C
Office Location:	Halsey 216
Office Hours:	Tuesday: 4:10pm - 5:10pm Wednesday: 10:30am - 12:30pm Thursday: 9:30am - 11:30am
Prerequisites:	Mathematics 103 with a grade of C or better, or qualifying for either Mathematics 104 or Mathematics 171 via Math Placement Test
Course Website:	http://www.uwosh.edu/d2l
Recommended Textbook:	A Balanced Introduction to Computer Science, Third Edition, David Reed. ISBN: 978-0132166751

Course Information

In this course, students will be introduced to the science of computing. The history of Computer Science, Elementary computer organization, big ideas in computing, algorithmic problem solving, and introductory programming concepts will be discussed. Programming concepts include data types, expressions, input/output, conditional statements, repetition, data processing, procedures and basic object-oriented principles. Programming concepts will be taught through the use of a specific programming language. The course assumes no previous programming experience.

Course Website

You should check d2l on a regular basis - it will contain lecture notes, handouts, assignments, announcements, and grades. Ill do my best to let you know when something new and important comes up, but it is your responsibility to check the web site frequently for information that you might not get otherwise.

Projects & Labs

Most projects and labs will consist of short programming projects. One of your goals (during this class and beyond, in Javascript or any programming language) should be to write understandable, readable code. You should be making every effort to comment anything that might be confusing to a reader unfamiliar with your program, to name variables intelligently, to use indentation that reflects the codes organization, and so on. All of this will be taken

into account during grading: poorly organized or written code may have a negative impact on your grade, even if the resulting program works fine.

Keep this in mind when writing programs: write your code in small pieces, making sure each piece works before moving on to the next one. It is much better to turn in a project that is not finished but has many working pieces than to turn in one that doesn't work at all, even though most of the code is written.

All assignments must be submitted electronically via d2l. It is your responsibility to ensure that your submission was submitted correctly. You must double check to ensure your program was uploaded correctly. **There are no late submissions.**

Resubmissions

Each assignment must pass **all** of my test cases. A program that fails just one test case will receive at most 70%. That said, you will be allowed to resubmit assignments *once* for regrading within 7 days of the initial program being graded. Complete test cases will be posted after the initial grading so you can go back and test your code to see what went wrong. Regraded programs can receive a maximum score of 85% if the modifications made were very minor. Significant modifications or many additions will receive very little, if any, additional points.

Exams

Exam material will come from the lecture, notes, labs and assignments. There will be more information about each exam as it approaches. The *tentative* exam dates are listed below. All exams will be taken during the regular class period. These may change, so as the date approaches make sure you've got the most recent information.

- **Exam One** - Tuesday, October 9th, 2018
- **Exam Two** - Tuesday, November 13th, 2018
- **Exam Three** - Tuesday, December 11th, 2018

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:

1. Make arrangements prior to the scheduled exam. For last minute emergencies, telephone me at 424-7080 or leave a message at the computer science office, 424-2068 or send me a text message. No after-the-fact notifications will be accepted.
2. Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

If allowed, only one make-up exam will be given. It will be a comprehensive exam given at an arranged time during the last week of the semester.

Grading

Course grades will be based on projects, labs and exams. Your final grade will be computed with the following percentages:

- 15% - projects
- 40% - labs
- 45% - exams

If you believe anything was graded incorrectly or unfairly and would like to have it regraded, you must let me know about it within *one week* of having the item graded. I will regrade the entire assignment or exam and you may gain or lose points.

Grading will be on a plus/minus system. Grading may be done on a curve depending on the overall performance of the class. If no curve is used, your grade will be computed based on the following:

Percentage	Grade	Percentage	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

Academic Dishonesty

Academic dishonesty of any kind will not be tolerated. All assignments, labs, mini assignments and exams are to be completed individually. While discussion of ideas and problems with fellow students is encouraged, all projects and labs must be done individually. In certain circumstances, code fragments from the instructor may be provided to eliminate tedious coding or to provide a common framework for all students. **All other code must be original.** Online resources may be used to help you understand the material, but you may not copy online code nor can you “borrow” code from other students, past or present.

Any suspected academic dishonesty will be dealt with on a case-by-case basis. Any clarification of what does or does not constitute academic dishonesty must take place **before** you turn in questionable work. For clarification on what constitutes academic dishonesty, contact me or consult the printed policy in the UWO Student Discipline Code, Chapter UWS 14.

Course Outcomes

1. Identify the core areas and big ideas in computer science.
2. Describe the difference between an algorithm and a computer program.
3. Identify the main components of the “von Neumann architecture.”
4. Describe the architecture and major components of a web application in a client server environment.
5. Given a description of a problem, apply the problem-solving steps used in computer programming to create a solution design.
6. Working from a solution design, implement a solution to a problem as a JavaScript program that runs in a web browser such as Firefox or Chrome.
7. Select the appropriate data types when implementing a solution to a problem using JavaScript.
8. Identify and implement selection control structures using if, if-else, and if-else-if statements in JavaScript.
9. Identify and implement repetition control structures using loops in JavaScript.
10. Identify and implement functions in JavaScript.
11. Design an HTML form for collecting user input with validation in JavaScript.
12. Process an array or collection of items in JavaScript.