

CS 262: Object-oriented Programming II

Term: Fall 2017
Credits: 4
Lectures: 10:20 AM – 11:20 AM, MTW in HS 457
Lab: 10:20 AM – 11:20 AM, F in HS 101C
Prerequisites: Math-108 or equivalent with a grade of C or better, or qualifying for a higher level mathematics via the Mathematics Placement Test, and CS 221 or equivalent with a grade of C or better.

Instructor

Instructor: Scott Summers
Email: summerss@uwosh.edu
Phone: 920-424-1324
Office: Halsey 220
Office hours: (subject to change):

	Mon	Tues	Wed	Thu	Fri
9:10 AM					
10:20					
11:30					
1:00					
2:00					
3:00					

Course description

A second course in problem solving, software design, and computer programming using an object-oriented language. Problem solving/software design topics include: abstract data types, universal modeling language (UML), simple recursion, unit testing, event-handling, simple concurrency. Data structures and algorithms include: binary search, simple sorting algorithms, use of collection classes and their iteration protocols, sequential file processing. Additional topics include: inheritance, polymorphism, graphical user interfaces, simple use of threads.

Course Website

The course website is: <http://www.uwosh.edu/d21>. You should check d2l on a regular basis, perhaps two or three times per day.

Required Textbook

Introduction to Java Programming, Comprehensive Version 10-th edition by Liang. ISBN 9780133761313.

We will cover material from chapters 8, 10, 11, 12, 13, 14, 15, 18, 19, 20, 21 and 30 – not necessarily in that order.

Course Grade

Your final course grade will be based on the following components.

20% HOMEWORK ASSIGNMENTS

There will be five programming assignments. All programs will be equally-weighted.

10% QUIZZES

You will be given at least one quiz potentially every week. Quizzes will be taken at the end of class. Each quiz is equally-weighted. Calculators will not be allowed for any of the quizzes. Quiz material will come from the lecture notes, labs, textbook and homework assignments.

10% LABS

There will be weekly labs, except during Thanksgiving week and during exam weeks. All labs are equally-weighted.

60% EXAMS

There will be three equally-weighted in-class exams. Exam material will come from the lecture notes, quizzes, textbook, programming assignments and labs. There will be more information about each exam as it approaches. The actual exam dates will be announced in class at least one week before the exam. All exams will be taken during the regular class period.

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, call me at 920-424-1324 or leave a message at the Computer Science office, 920-424-2068. **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 Scale

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, then your grade will be computed based on the following:

Percentage	Grade
> 91	A
> 89 and \leq 91	A-
> 87 and \leq 89	B+
> 81 and \leq 87	B
> 79 and \leq 81	B-
> 77 and \leq 79	C+
> 71 and \leq 77	C
> 69 and \leq 71	C-
> 67 and \leq 69	D+
> 61 and \leq 67	D
> 55 and \leq 61	D-
\leq 55	F

Re-grading

If you believe an assignment, quiz, lab or exam was graded incorrectly or unfairly and would like to have it re-graded, please let me know about it in writing within one week of receiving the assignment, quiz, lab or exam back. I will re-grade the entire assignment, quiz, lab or exam and you may gain or lose points.

Late Work

Late work will NOT be accepted. Late work is worth 0 points. Extensions 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.

Coding Standards

In this class, you will write several short to medium-length Java programs. One of your goals (during this class and beyond, in Java or any programming language) should be to write understandable, readable code. You should be making every effort to document anything that might be confusing to a reader unfamiliar with your program using correct spelling and grammar, to name variables intelligently, to use indentation that reflects the code's 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. Briefly, your code must be formatted in a consistent and easily-readable manner. At a minimum, I will require that you utilize the auto-format feature of whatever development environment we use (e.g., "Auto-layout" in BlueJ).

One of the goals of this class is to teach you to write functioning programs in Java. Thus, your code must compile and run correctly in order for you to receive full credit. Code that does not compile will receive substantially less than full credit. 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 and labs must be submitted electronically via d2l (each lab and assignment will contain specific instructions). It is your responsibility to ensure that your assignment or lab was submitted correctly. You must double check to ensure your assignment or lab was uploaded correctly.

University Policy on Academic Integrity

The University of Wisconsin Oshkosh is committed to a standard of academic integrity for all students. The system guidelines state: "Students are responsible for the honest completion and representation of their work, for the appropriate citation of source, and for respect of others' academic endeavors" (UWS 14.01, Wisconsin Administrative Code).

Often, students are not aware of the ways to identify and avoid plagiarism. Therefore, it is important to educate yourself about how to give proper credit to sources that you use in your assignments. For writing assignments, you can consult the Purdue Owl website on how to identify and avoid plagiarism: <https://owl.english.purdue.edu/owl/resource/589/02/> and <https://owl.english.purdue.edu/owl/resource/589/03/>. This website outlines the strategies for avoiding plagiarism in this course. However, other courses may demand knowing other ways to identify and avoid plagiarism. Therefore, I encourage you to consult with me if it is unclear to you how you give proper credit to your sources of information.

In sum, all material turned in for this course must be original. In this course, you may not re-use papers or projects from other sections of this course, from other courses you have completed, or other courses you are currently completing. This class is a specific event in your learning process. To learn, you must engage in the material and complete the work. Thus, work from other experiences is not acceptable. All work turned in that is plagiarized will receive a "0" in the course.

Course Objectives

At conclusion of the course, students will be able to:

1. Debugging Java programs with BlueJ – You will be expected to:
 - (a) Analyze a program on its correctness
 - (b) Identify software bugs with the debugger in BlueJ
2. Objects and Classes – You will be expected to:
 - (a) Specify a class with the UML graphical notation
 - (b) Use the UML graphical notation to describe classes
 - (c) Distinguish between object reference and primitive data type variables
 - (d) Apply classes in the Java API (Application Programming Interface)
 - (e) Differentiate between instance and static variables
 - (f) Develop methods in classes
 - (g) Store and process objects in arrays
 - (h) Apply class abstraction to develop software
3. Inheritance and Polymorphism – You will be expected to:
 - (a) Develop a subclass from a superclass through inheritance
 - (b) Apply the polymorphism concept to handle different data types using a uniform interface
4. Abstract Classes and Interfaces – You will be expected to:
 - (a) Identify the similarities and differences between an abstract class and an interface
 - (b) Model weak inheritance relationships with interfaces
 - (c) Specify a natural order using the Comparable interface
 - (d) To wrap primitive data values into objects
 - (e) Create a generic sort method
 - (f) Simplify programming using JDK 1.5 automatic conversion between types and wrapper class types
5. Exceptions and Assertions – You will be expected to:
 - (a) Distinguish exception types: Error version Exception in Java.
 - (b) Throw an exception in a method
 - (c) Write an exception handler using a try-catch-finally block
 - (d) Explain the propagation of an exception
 - (e) Apply assertions to help ensure program correctness
6. Text I/O – You will be expected to:
 - (a) Read and write characters using the `InputStreamReader`, `FileReader`, `BufferedReader`, `OutputStreamWriter`, `FileWriter`, `PrintWriter` and `BufferedWriter` classes.
 - (b) Be able to apply the appropriate class in text I/O operations based on the requirements and performance needs
 - (c) Distinguish between text I/O and binary I/O
7. Object-Oriented Design – You will be expected to:

- (a) Become familiar with the software development process
 - (b) Model a system with the appropriate relationships: association, aggregation, composition, dependency, strong inheritance and weak inheritance
 - (c) Declare classes to represent the relationships among them
 - (d) Design systems by identifying the classes and discovering the relationships among these classes
8. Unit testing with JUnit – You will be expected to:
- (a) Create test classes, test methods and run tests with JUnit
 - (b) Create and use test fixtures in JUnit
 - (c) Interpret test results with JUnit
 - (d) Correlate the test fixtures with assertions
 - (e) Verify that a software unit performs as specified
9. GUI and Graphics – You will be expected to:
- (a) Describe the Java GUI hierarchy
 - (b) Create user interfaces using frames, panels and simple GUI components
 - (c) Apply layout managers
 - (d) Use `JPanel` as sub-containers.
 - (e) Draw figures using the methods in the `Graphics` class.
 - (f) Override the `paintComponent` method to draw figures on a GUI component
 - (g) Introduction to Threads – creation, simple usage
10. Event Driven Programming – You will be expected to:
- (a) Declare listener classes and write event handlers to handle events
 - (b) Apply the Observer pattern to decoupled programs
 - (c) Register listener objects in the source object
 - (d) Create inner classes and anonymous inner classes
 - (e) Write programs to handle `ActionEvent`, `MouseEvent`, `KeyEvent` and `Timer` events.
11. Recursion – You will be expected to:
- (a) Solve problems with recursion
 - (b) Write programs using recursion
 - (c) Explain the difference between iteration and recursion
12. Generic Types – You will be expected to: Improve reliability and readability of Java programs by using generic types
13. Java Collections Framework – You will be expected to:
- (a) Describe the Java Collections framework hierarchy
 - (b) Utilize the common methods in the `Collection` interface for operation sets and lists
 - (c) Utilize the `Iterator` interface to traverse a collection
 - (d) Examine the `Set` interface and be capable of deciding when to use `HashSet`, `LinkedHashSet` or `TreeSet` to store elements
 - (e) Compare elements using the `Comparator` interface
 - (f) Examine the `List` interface and be capable of deciding how and when to use `ArrayList` or `LinkedList` to store elements
 - (g) Examine the `Collection` and `Map` and be capable of deciding how and when to use `HashMap`, `LinkedHashMap` and `TreeMap` to store values associated with keys.

Students' Rights and Responsibilities: Course Policies

WELCOME STUDENTS!

As a University of Wisconsin–Oshkosh student, you have rights and responsibilities regarding your relationship with both the classroom and university community. The policies in the course specifically as well as at UW–Oshkosh generally are in place to foster an equitable and safe classroom and campus climate. The primary goal of all policies is to create a classroom and campus community in which all students can access learning, achieve success and reach their goals. Your rights and responsibilities are important to know both so that you can enhance your learning experience and contribute positively to the campus climate. Specific policies and procedures can be accessed through the Student Affairs Policies and Procedures webpage (<https://www.uwosh.edu/stuaff/policies-procedures>).

INCLUSION STATEMENT

Building relationships and community is one of the most important goals of the course. The only way to build community in this course is to honor each person in terms of their identity. Each student in the course will conceive of their identity in different ways; aspects of identity important to students in the course may include race, ethnicity, ability, sex, sexuality, gender, gender expression, gender identity, religious beliefs, political affiliations, and/or class. Thus, each of us, myself included, must honor each students' identity in all its complexity. We need to work on taking up perspectives unlike our own, challenging our assumptions and finding a route toward understanding the similarities and differences between ourselves and others.

STUDENTS WITH DISABILITIES

Most importantly, students with disabilities are welcome in this course! If you need alternative/additional instructional structure for this class due to specific individual learning needs, please talk to me and we can work together. I am committed to creating an environment conducive to learning for all students.

UNIVERSITY POLICY ON TYPES OF EXCUSED ABSENCES

There are several forms of absences that are excused under University policy: “Students are excused from class for participation in all-University events [GEN 4.B.10 (1)(b)] and for circumstances beyond the students' control including, but not limited to medical or family emergencies (medical care for pregnancy, illness, child care issues, death or serious health problem of family member), court appearance, required military service not to exceed two (2) weeks unless special permission is granted by the instructor or chair, jury duty, etc.” Student responsibility: “Students are responsible for notifying the instructor. . . as far in advance as possible and may not be penalized for such absences as long as appropriate documentation is provided in a timely fashion to the instructor to verify the reason for the absence.” Instructor responsibility: “Instructors are responsible for providing reasonable accommodation or opportunities to make up course obligations that have an impact on the course grade.”

RELIGIOUS ACCOMMODATION FOR STUDENTS

Both University policy and state policy requires that instructors honor students sincerely held religious and faith traditions by making accommodations for religious holidays or other days of special religious significance. If there is a scheduling conflict for you between attending a course session and/or completing coursework on a day of religious observance, then, it is necessary to “notify the instructor within the first three weeks of the beginning of classes of the specific days or dates on which you will request relief from an examination or academic requirement.”

CREATING A SPACE FREE FROM SEXUAL HARASSMENT

The University policy on sexual harassment is very clear: it will not be tolerated anywhere on campus, including the classroom. Sexual harassment is defined by the University of Wisconsin Oshkosh as follows: “Sexual harassment is a form of sex discrimination. It [is] . . . the inappropriate introduction of sexual activities or comments into the work, learning, or living situation. Such behavior is not acceptable at the UW–Oshkosh and will result in disciplinary action.”

ATTENDANCE

I am dedicated to your success and know that attendance is crucial to achieving improvement in your skills and abilities and, thus, your success in the course. Therefore, I may keep track of attendance weekly. Missing two weeks or more of the course may result in a failing final course grade. In turn, if I receive no communication from you for two weeks, I reserve the right to withdraw you from the course.

If you experience difficulties such as illness or death in the family or other significant disruptions in your life as discussed in the section on excused absences above, then, please communicate with me about your situation and we will forge a plan on how to best catch you up in the course.

DROPBOX

Odd things happen in cyberspace—emails get lost, servers disconnect temporarily, and logins fail. Due to this challenge, you should anticipate possible mishaps and complete your work with enough time to meet the deadline. In turn, timely communication aids success: reply to emails received and check for replies to your sent emails. With these strategies, you will be able to meet my expectations of getting work in on time.

Policy on Electronic Cigarettes

The use of electronic cigarettes (e-cigarettes) of any kind within the classroom is strictly prohibited.