CS 145 Computer Science II

Spring 2022 Section -003 JHSW-216 (MoTuWeTh 10:10AM - 11:05AM)

Instructor: El Magrous, Ahmed, Ph.D.

Office: 234E Jarvis Hall Science Wing
Phone: 715-232-1946
E-mail: elmagrousa@uwstout.edu
Office Hours: Mon, Tue, Wed, Thur 1:15-20:30 p.m.
Mon, Tue, Wed, Thur 3:30-4:15 p.m.

Course Description

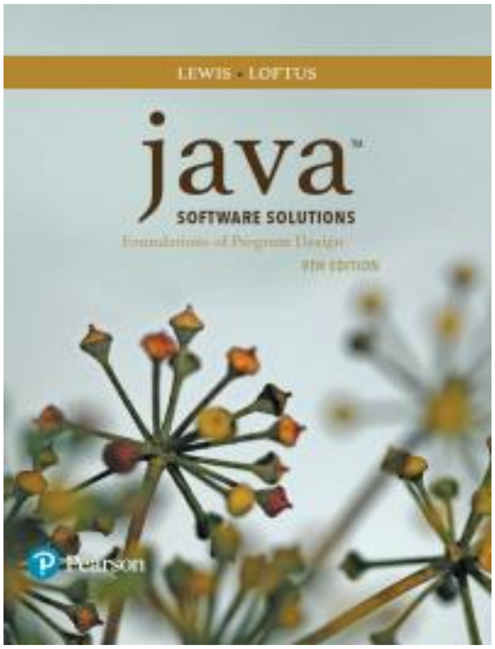
This course introduces students to object-oriented design and programming using Java via both classroom and laboratory work. The course provides the fundamental concepts of programming from an object-oriented perspective including abstract data types, encapsulation, inheritance, and polymorphism. The course emphasizes developing fundamental programming skills in the context of a language that supports the object-oriented paradigm (Java). The students will learn to design algorithms in the OOP style and to translate them into correct programs as well as to debug, document, test, and maintain the code. The Java constructs covered include: conditional and loop statements, functions, arrays, object classes, files, interfaces, inheritance, exception handling, Graphical User Interface (GUI), and windows forms (FX).

In-class lab exercises and weekly programming assignments will help students to practice using the course concepts to solve problems. All assignments and class resources will be available on Canvas. The course is expected to be challenging in terms of workload and difficulty. You are expected to be prepared to do a significant amount of work (8-10 hours per week). The good news is that you will be learning a lot of useful and interesting programming concepts that will help you succeed in your computing life in both academia and industry.

Course Objectives

Upon successful completion of the course, students will be able to

- Demonstrate understanding of object-oriented programming including abstract data types, encapsulation, inheritance, and polymorphism.
- Analyze the relationship between classes and objects.
- Apply testing and documentation of classes.
- Apply and understand exception handling in programming.
- Create Graphical User Interfaces (GUIs) and implement event handling.
- Create software systems in the Java programming language.



Textbook and Supplements

Java Software Solutions, 9th Edition

- by John Lewis William Loftus
- Publisher: Pearson;
- ISBN-13: 978-0134462028.

Reference Books (optional)

Building Java Programs, 4/E or 3/E

- By Walter J. Savitch, Pearson,
- 2014, 10TH Edition

The Java™ Tutorials (Oracle)

- https://docs.oracle.com/javase/tutorial/java/concepts/

Eclipse: Total Beginner Lessons (how to use Eclipse for Java)

- https://help.eclipse.org/2019-12/index.jsp?nav=%2F0
- http://eclipsetutorial.sourceforge.net/totalbeginnerlessons.html

Course Resources

This class will be conducted using Canvas, a course management tool. I will place quizzes, assignments, lecture slides, and other course materials in the order that they should be completed. If you are absent, please go to the course home page and download the appropriate materials. Your participation in the class discussion will be worth some points (see grade distribution).

Software

Eclipse (Free Edition) will be used to create programs. You need to download and install it as soon as possible.

➢ Quick Guide:

1) Download and install **JDK** (up to date release, e.g., 13.0.2).
   https://www.oracle.com/technetwork/java/javase/downloads/jdk13-downloads-5672538.html
2) Use the eclipse installer to help you install Eclipse easily.
   https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2019-12/R/eclipse-inst-win64.exe

Note: If you still need help, see the detailed guide posted on Canvas. Of course, you are welcome to meet with the teaching assistant or me.

Supplemental Instruction

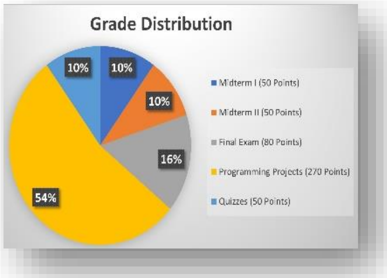
There is a supplemental instructor (SI) assigned to each section, **Luke Linna (linnal0108@my.uwstout.edu)**. Your SI leader will sit in on all lectures and be available for help during the discussion and hands-on exercises. ***Luke*** will also be responsible for checking off your weekly homework. SI leaders are also available outside of class in the CS tutor lab. You should plan on attending the SI at least once a week. Students who get help from the SI in the tutoring lab on a regular basis typically score one letter grade higher than those who do not. These are supplemental to class and highly recommended. They are an excellent opportunity to review class materials, get questions answered, get help on assignments, etc. Please make good use of this resource.

Virtual lab

The virtual lab is through canvas https://uwsto.instructure.com/courses/390729 as a recurring meeting in Zoom https://wisconsin-edu.zoom.us/j/99556859940. The Zoom meeting is setup with a Waiting Room, so tutors can use that as a queue. SI have host privileges, they can also add breakout rooms if they need to.

Assessment and Grading (Organization)

A significant portion of the material for this course will be presented only in lectures, so students are expected to attend lectures regularly. There are 500 points (approx) you can earn toward a grade in the course, and your performance will be measured in multiple ways. (The percentage contributions to your grade given below are approximate and subject to change.) Grades will be available via your student account on **CANVAS**.



10%	50 points, In-Class **Midterm 1 Exam**,(tentative) Thursday, Feb 24, 10:10AM – 11:05AM
10%	50 points, In-Class **Midterm Midterm 2 Exam**, (tentative) Thursday, April 14th, 10:10AM – 11:05AM
16%	80 points, In-Class Written **Final Exam**, (Confirmed) Friday, May 6, 8:00 - 9:50 AM
54%	270 points, roughly 9 **Programming Projects/homework**.
10%	50 points, roughly 10 **quizzes** (via Canvas), 5 points each

The sum of the possible scores on all evaluation components is considered 100% and your final course grade will be determined as follows:

Scale	0%	60%	70%	73%	78%	80%	83%	88%	90%	93%
Grade	F	D	C-	C	C+	B-	B	B+	A-	A

There will be no curve at the end of the course. Your score will not be rounded up: if you get 69.99%, you will get a D, not a C-. Thus, you should always be able to determine how well you are doing in the course.

Weekly Programming Assignments (homework)

The lectures are complemented by programming work and assignments. Programming assignments are an integral part of the course, and those assignments constitute a significant portion of the course grade. The programming work policies are stated on the course website and are to be followed for the success of the learning process.

The programming tasks' grades will also be derived from completing assignments within due dates. The programming tasks for CS 145 is a sequence of relatively short programming assignments (~9). Each assignment is worth 30 points (~270 total), and all are equally weighted and may consist of multiple parts. The programming assignments must be submitted into designated drop boxes via CANVAS for instructor grading. It is recommended that you keep in touch with your instructor and ask for assistance promptly when needed.

Notes:

- ✓ *Your work for the programming assignments must be checked off the SI (*Luke Linna*).*
- ✓ *Late homework assignments will not be accepted. So, no extensions will be granted. Why?! Solutions will be discussed (not posted) right after the due date.*
- ✓ *Discuss unusual circumstances in advance with the instructor. I will always be willing to assist.*

Quizzes

There will be (roughly) 10 quizzes that need to be taken online via Canvas by the due date. Each Quiz is worth 5 points (50 in total). CANVAS quizzes are designed to make you read the materials. These are open book short tests (~10-15 Minutes). However, you need to be well prepared in order to complete it on time and do well. A guide will be posted prior to the quiz due date. Missed quizzes cannot be made up.

Midterm (2) and Final Written Exams

There will be two online via canvas midterm exams each worth 50 points, and a final exam worth 80 points. Exams will consist of two parts: concepts and programming. Guide and sample questions will be posted on CANVAS and discussed in class at the right time.

Class Participation (in lecture and (Links to an external site.)discussion group)

Class attendance is an essential component for understanding the concepts of the topics due participating in class discussions, hands-on exercises, and potential extra homework.

Attendance

Attendance is expected of all students for every class. That includes being on time for class and staying the allotted time. Lecture attendance is the most critical component of the course and is reflected in the final grade. Additionally, it is highly recommended that you keep in touch and reach out to your instructor as needed. Following the course plan and reading and completing the assigned work in a timely manner is the most critical component of the course and is reflected in the final grade. You are very important to the discussion and group activities.

Lab Programming Exercises

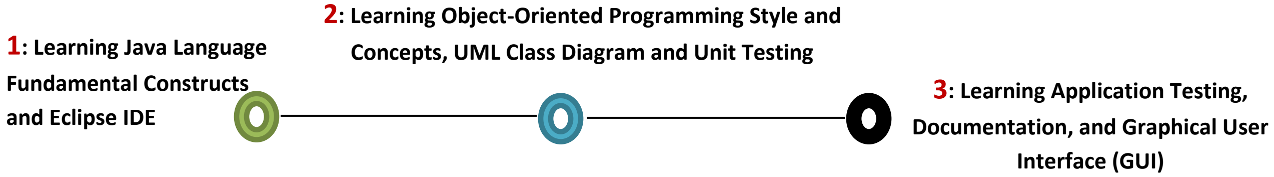
There will be expected (preferably daily) lab programming (hands-on) exercises that are expected to be conducted in class. The idea is to apply, hands-on, all the stuff covered in class. They will not be graded; however, they will be considered as positive class participation. Again, please do not be confused with the homework programming assignments that are going to be graded and expected to be submitted by the due dates via CANVAS.

Student Responsibilities – Take control of your learning!

1) You are expected to complete the assigned reading with a reasonable level of understanding. Please keep in touch and reach out to your instructor as soon as help is needed.
2) Class lecture notes are provided If you desire a printed copy, it is your responsibility to print them yourself. Just make sure that you never share them with the public (copyrights).
3) You are expected to complete in-class activities by the next class period as some of the activities will build on each other.
4) You are expected to have access to a computer needed for working on the programming assignments, taking the quizzes, and working on the extra exercises if time allows.
5) I may send info to you by Canvas Course news items and campus email, so you should check them frequently.
6) You are important to the class discussion, your participation is expected.
7) Don't worry as much about grading and points as you do about learning.
8) Do your work and try not to fall behind in assignments.
9) The assignments tend to be time-consuming. Don't wait until the last minute to begin them! You should plan on devoting approximately 8-10 hours of work per week. Of course, I will always be happy to assist.
10) Use MS-Teams. It is a great tool to communicate with your class-fellows and me.
11) Emails are filtered so you must include the prefix (CS145-003) to your email subject.

Course Phases (my plan)

To make the course well organized and effective, I will be dividing it into three main phases. Each phase will include some goals to achieve, hopefully, in a timely manner.


1: Learning Java Language Fundamental Constructs and Eclipse IDE
2: Learning Object-Oriented Programming Style and Concepts, UML Class Diagram and Unit Testing
3: Learning Application Testing, Documentation, and Graphical User Interface (GUI)

Phase 1: Learning Java Programming Language and Eclipse IDE

In this phase, we will first be working on learning Java fundamental constructs and libraries (packages). In every class meeting, you will learn and be exposed to new Java predefined classes and methods. In-class hands-on lab exercises will help you practice debugging programming in Java projects using Eclipse IDE.

Phase 2: Object-Oriented Programming Style, UML Class Diagram, and Unit Testing

In this phase, we will start learning and using the OOP style. I will be covering the main OOP concepts and principles (e.g., Abstract Datatypes, Modularity, Reusability, Encapsulation, Polymorphism, Interfaces, and Abstract Classes). We will be using UML Class Diagram in designing our programs and determining the application classes and relation between them. Additionally, you will be learning about software testing. We are going to focus on unit testing in this phase.

Phase 3: Software Testing, Documentation, and Graphical User Interface

In this phase, you will be learning more about program testing and documentation (JavaDoc). You will be working on larger-scale programs and will involve group projects where the work is divided among a programming team. Teams will be formed and posted on CANVAS. Near the end of the semester, we will be studying a new GUI library (Java FX). You will learn how to design windows forms and work with event-driven programming.

Notifications in Canvas:

CanvasNotifications.pdf

Student Help Guides:

Link: https://community.canvaslms.com/docs/DOC-10701

Canvas Student Help Guides are available to online students to get information on "Getting Started," "Course Planning," "Collaboration and Communication," "Creating Conferences," and "Assessments." More specifically, the guides can be used as job aids to help you to navigate the News, Content, Discussions, Dropbox, Grades, Quizzes, and Checklist features of the course site.

Technical Assistance:

Link: https://kb.uwstout.edu/page.php?id=81944

The Technical Assistance link will lead you to a web page with information on where to get help with Canvas. Click on the link "Canvas 24/7 365 Support." You can get help from a Canvas expert, report a problem, and find answers to common questions. You're also able to submit an idea to improve Canvas.

FAQ's:

Link: https://community.canvaslms.com/community/answers

The Frequently Asked Questions page is where online students will find common Canvas questions/answers on password issues, submission issues, adding/dropping courses (in relationship to unenrollment from a course site), and recommended browsers.

Tech HelpDesk:

Link: https://kb.uwstout.edu/

Online students can also interact with the Technology Help Desk, here. This link will lead you directly to the UW-Stout Knowledgebase. You can also create a help/service ticket, here, as well. Further, view Open Hours at this location, and get Password Assistance on this page.

E-Textbook Support:

Link: https://liveuwstout.sharepoint.com/sites/2022/029/Pages/Textbooks/E-Textbook-Support.aspx

Need information on E-Textbooks? When using RedShelf products, online students can view Frequently Asked Questions and user guides, here. There's also information on this page for Pearson MyLab, regarding student registration and support. For help with any of these areas, please contact the Digital Resources Team.

Disability Services

UW-Stout strives for an inclusive learning environment. If you anticipate or experience any barriers related to the format or requirements of this course, please meet with me so that we can discuss ways to ensure full access. If you determine that additional disability-related accommodations are necessary please contact the Disability Services office (Room 120 in the Robert S. Swanson Library and Learning Center, ext. 2995, http://www.uwstout.edu/disability).

Academic Integrity and Misconduct

Students are responsible for the honest completion and representation of their work, for the appropriate citation of sources, and for the respect of others' academic endeavors. Students who violate these standards must be confronted and must accept the consequences of their actions.