CSE114 : Introduction to Object-Oriented Programming

Syllabus

Term: Spring 2025
Instructor: Tony Mione
Office: [B425]
Phone: +82 032-626-1226
Email: antonino.mione@sunykorea.ac.kr
Course Meeting Times: Lec: Mon & Wed, 2:00-3:20 PM
Lab: Tues & Thurs, 2:00-3:20 PM
Office Hours:
Mon/Wed: 10:30-12:00 AM
Tue/Wed: 1:00-2:00 PM
Thur: 11:00-12:00 PM
(or by appointment) [B425]
Course Homepage: https://www3.cs.stonybrook.edu/~amione/CSE114_Course/index.html
Brightspace:
Lecture: https://mycourses.stonybrook.edu/d2l/home/1802351
Lab: https://mycourses.stonybrook.edu/d2l/home/1802361
Required Texts:

- Downey, Allen B., Mayfield, Chris, "Think Java: How to think Like a Computer Scientist", 2nd Edition, Green Tea Press, 2017

Recommended:

- Liang, Daniel, "Introduction to Java Programming", Brief Version 10th Edition, Prentice-Hall, 2014, (ISBN-10: 0-13-359220-0; ISBN-13: 978-0-13-359220-7)
- Browse the Java SE 8 or Java SE 21 API Specification. (Either is fine. SE 8 page is older but easier to read. It would not matter for us in CSE 114.)
- **The Java Programming Language, 4th edition**, Arnold, Gosling, and Holmes, Addison-Wesley.

Course Organization

The focus of the course will be on teaching you how to begin with a problem statement and then systematically design a computer program that solves the problem. The idea is to give you the proper knowledge, intuition, and habits on which you can base a lifetime of learning in computer science.

You will be using a subset of the programming language Java, the core subset, using a programming environment called Intellij IDEA. (Do *not* use Intellij IDEA until we are ready to use it though. I will let you know when that is.) Intellij IDEA is designed for the programming professionals and is widely used in both academia and industry. It is most likely that you will be using a small subset of its features in this course, but you are welcome to explore as much as you wish as you program in Java this semester and beyond.

Course Objectives/Outcomes

Upon completion of the courses, students are expected to possess:

- An ability to program in an object oriented language, using concepts such as object classes, encapsulation, inheritance, and polymorphism.
- An ability to use fundamental data structures such as arrays.
- An ability to program with sound code structure and use systematic software debugging and testing techniques.

Prerequisites

- Level 4 or higher on the math placement exam
- Advisory Prerequisite: CSE 101 ➔ Important note: if you have no prior programming experience (i.e., have not taken a course equivalent to CSE101, we **STRONGLY** recommend you switch to CSE101 as that course will be more manageable than CSE114 and will prepare you to succeed in this class in an upcoming semester.

Major Topics Covered in Course

- Programming environment including a text editor, command-line interface, and an IDE, e.g., JetBrains Intellij IDEA.
- Functions including parameter passing mechanisms. Introduction to formal methods (pre-conditions, post-conditions, loop invariants)
- Control structures (conditionals and loops).
- Arrays.
- Objects using predefined classes, e.g., String.
- Writing moderate to fairly complex user-defined classes.
- Interface, inheritance, polymorphism, abstract classes. The Java class hierarchy.
- Exceptions and file I/O.
- Recursive programming.
- The ArrayList class.

Attendance

The range of topics covered in this course is extensive, and due to the limited lecture and lab time, these topics are covered in an intensive manner. Therefore, attendance at both lectures and labs are mandatory in order to keep up and perform well.

- Attendance will be taken at each lecture and lab session.
- A sheet is passed around that must be signed
    - ONLY Sign for yourself! Signing for anyone else is fraud and is a breach of academic integrity. If caught it will be reported!
    - **Make sure you sign the sheet during class! I cannot mark you present based on an email after class saying "I was at class and forgot to sign. Please mark me present."**
- ***If a student has over 20% unexcused absences, the final course grade will be an F.***

Grades and Evaluation

Your grade in the course will be based on the following work:

Attendance/Participation – 5% – Based primarily on attendance at class AND labs.
Assignments – 50% – Programming assignments will be given each week including some problems from the text book that will give the student an opportunity to apply the knowledge acquired from reading and lectures.
Labs – *Labs, in general, are for deeper exploration and are not graded. However, 3 or 4 times during the year, I will ask that you submit your work from the lab. I will tell you after half the lab class is finished if I want something submitted (please do not ask me at the start of the lab class). These submitted labs are a 'spot check' for me to assure the students are doing lab work. The grades given for those are rolled into the assignment grades.*
Exams – 30% – 3 exams spaced at about 1 month intervals (10% each). The exams are based on reading, lecture slides, and demos presented in class and test ability to write short programs and understand Java syntax and semantics.
Final Exam – 15% – A Cumulative final exam will provide questions that will cover the key concepts taught during the entire semester.

Final Grade Calculation

Important note: ***You must attain a grade of at least 60% on exams and 60% on problem sets to achieve a grade higher than C-. If you score under 60% on either the exam average or the homework average, you will receive a C- or lower grade even if your overall grade is above 60%!***

Letter grades are given on the following scale:

Letter	Minimum Percentage
A	90
A-	85
B+	80
B	75
B-	70
C+	65
C	60
C-	57
D+	54
D	50
F	<50

Course Schedule

Following is a tentative schedule for the class topics:

Week/Day	Lec	Lecture Topics	Readings	Tests/Assignments
W1: 2/24 2/26	1 2	Course Overview, Elementary Programming Programming Overview	Chapter 1	***[Assign dates approximate]***
2/25, 2/27 [Labs]		Installing Java, Using Java API Doc website		
W2: 3/3 3/5	 3	Subs: Ind Movement Day: No classes Program Structure, methods, variables,	 Chapters 2, 3, 4	
3/4, 3/6 [Labs]				
W3: 3/10 3/12	4 5	Types, Functions and Conditionals More functions and conditionals	Chapters 2, 3, 4, 5	Assign1 Due
3/11, 3/13 [Labs]				
W4: 3/17 3/19	6 7	Iteration (while and for loops) Packages, information representation, Binary	 Chapter 6.1-6.4, 7	
3/18, 3/20				
W5: 3/24 3/26	8	Arrays and variable scope	Downey: Chapter 9.1, 9.2, 9.4	Assign2 Due
3/25, 3/27 [Labs]		Exam I Review [924]	[Lec 1-6]	Exam I [3/27]
W6: 3/31 4/2	9 10	Advanced Arrays, 2D Arrays Static Fields, Command line arguments	Downey: Chapter 9.1, 9.2, 9.4	Assign3 Due
4/1, 4/3				
W7: 4/7 4/9	11 12	Sorting and Searching Object References	Chapter 9, 10, 11, 12	Assign4 Due
4/8, 4/10 [Labs]				
W8: 4/14 4/16	13 10a	Array of Objects Assertions, Reasoning and Logic	Chapter 9, 10, 11, 12	Assign5 Due
4/15, 4/17				
W9: 4/21	14	Equality Testing, object passing [Lab 4/23]	Chapter 9, 10, 11, 12	Assign6 Due
4/22, 4/24		Exam II Review 4/22	[Lec 7-13]	Exam II [4/24]
W10: 10/28 10/30	15 16	Static/Dynamic members, visibility The '.' Operator, 'this' object references1	Chapter 9, 10, 11, 12	Assign7 Due
4/29, 5/1 [Labs]				
W11: 5/5 5/7	 17	Children's Day/Budda's BD : No class Exception handling, File I/O	Liang: Chapter 12 13.6	Assign8 Due
5/6, 5/8 [Labs]		Subs Children's Day/Budda's BD : No class Lab: Exc Handling/File I/O		
W12: 5/12 5/14	18 19	Comparing objects/object sorting Interfaces/interface inheritance	Chapter 14	Assign9 Due
5/13, 5/15 [Lab]				
W13: 5/19 5/21	20	Inheritance (implementation)		
5/120, 5/22 [Labs]		Exam III Review 5/20		Exam III [5/22]
W14: 5/26 5/28 5/30	21 22 23	Array Lists, Generic Classes Abstract Classes Recursion	Chapter 8	Assign10 Due
5/27, 5/29				
W15: 6/2		Review for Final		
W 16: 6/9		Final Exam [12:30 PM – 3:00 PM]		

Programming Assignments

Development Environment

Most of the early programming that you do in this course will be in Java built on the command line. IntelliJ will be used from about one month into the class.
Emacs, Java, and IntelliJ are all available free for you to install on your computer. If you don't have a computer of your own to use, please let me know as soon as possible.

Academic Integrity: Cooperation vs. Cheating

Working with others on assignments is a good way to learn the material and we encourage it. However, there are limits to the degree of cooperation that we will permit.

When working on programming assignments, you must work only with others whose understanding of the material is approximately equal to yours. In this situation, working together to find a good approach for solving a programming problem is cooperation; ***listening while someone dictates a solution or any lines of Java code is cheating***. You must limit collaboration to a ***high-level discussion*** of solution strategies, and stop short of actually writing down a group answer. Anything that you hand in, whether it is a written problem or a computer program, must be written in your own words. If you base your solution on any other written solution, you are cheating. Finally, ***if you use an AI to generate any part of your code, it is cheating***. If caught, this will be reported to the academic integrity committee.

It is okay to help other students, within limits. ***If you are asked for help by another student, two things that are absolutely forbidden are to show that student your solution or to put your hands on that student's keyboard or paper***. That isn't helping; that is facilitating cheating! Instead, answer questions; give tips; help with tools; explain Java; point out a bug; and/or give encouragement. In other words, interact with other students the way that the TAs do.

When taking an exam, you must work completely independently of everyone else. Any collaboration here, of course, is cheating.

We do not distinguish between cheaters who copy others' work and cheaters who allow their work to be copied.

If you cheat, you will be referred to the appropriate office at the University. If you have any questions about what constitutes cheating, please ask.

This is what the University says: Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. *Faculty members are required to report any **suspected** instances of academic dishonesty* to the Academic Judiciary Committee or the Department of Academic Affairs.

Re-grading

For re-grading of an assignment or exam, please meet with the person (instructor or teaching assistant) responsible for the grading. All such requests that are later than one week from the date the graded work is returned to the class will not be entertained. ***To promote consistency of grading, questions and concerns about grading of assignments should be addressed first to the grading TA and then, if that does not resolve the issue, to the instructor.*** You are welcome to contact the TA by email or come to his/her office hours. If you would like to speak with the TA in person, and have a schedule conflict with his/her office hours, you are welcome to make an appointment to meet the TA at another time. For exams, you may contact the instructor directly via private post on campuswire or by email.

Getting Help and Information

Campuswire

We will be using Campuswire, which is a forum website that you can post questions and get answers from TAs, the professor, and classmates.

The Campuswire discussion board should be used for all communication with the teaching staff for questions about the course assignments and material. Email should be sent to individual instructors or teaching assistants only to schedule appointments.

Campuswire is a forum for additional learning and assistance. The following are not appropriate uses of Campuswire:

- cyber-bullying
- posting memes
- complaining about a grade
- airing concerns/comments/criticisms about the course
- posting more than a few lines of source code from an attempt at a homework problem
- posting the solution to a homework problem or a link to a website containing the solution
- in general, anything unrelated to the course material and student learning

Therefore, students are expected to use the Campuswire forum for all non-personal, course-related communication. Questions about what a homework problem is asking, technical problems that need troubleshooting, or other questions that might be of interest to other students must be posted to Campuswire and not emailed to the instructor or a TA. However, you may email if your questions are related to logistic issues with your coursework or attendance.

If code is relevant to a student's Campuswire question, the student may post only short code snippets. For more extensive help with reviewing or debugging code, students must attend office hours.

I encourage you to see me when you need help, advice, or encouragement. I will always be available during my regular office hours each week, and you may also make appointments for other times.

How To Approach This Class

For most of you, this will be your first or second class in computer science. Here is some advice on how to approach this class.

- Skim the relevant chapter(s) in the suggested reading before you come to lecture, and read more carefully after the lecture. The lecture will not assume that you have read the suggested chapter in advance, and the lecture should help you understand the context and key points of the chapter. Nevertheless, the lecture does not fully replace reading, because the lecture may omit fine points that are discussed in the reading assignment.
- Concentrate in lecture. The concepts that are presented in lecture are what's important; you will be able to find the details in the book or in my lecture notes. Writing down everything that is said and then trying to figure it out later would not be a good way to learn. Instead, think about what is being said. I will ask many questions. Try to answer all of them even if you do only in your head. Raise your hand and ask a question when you don't understand something. Try to understand everything. Don't give up!
  You are welcome to ask as many questions as you wish, but try not to answer too many of my questions. This is to give others a chance to answer them too.
- Participate in the labs. You'll be sitting at a computer in a room with the instructor, a teaching assistant, and other students. Take advantage of the computer by trying things out. That way, you'll discover the things that you don't understand in a setting where there are plenty of others to help you out. Try to read the lab assignment before you come.
- Respect the assignments. Some students expect that the assignments will be straightforward if they have done the reading and concentrated in lecture. Not so! The assignments are designed to challenge you by requiring that you apply the concepts you have learned during lecture to new situations. *The assignments will be your most important learning experience in the course.* They will rarely be straightforward. First couple assignments may seem straightforward but that is not a good indication of what the remaining assignments will be like. Assignments become more complex as we incorporate more concepts and ideas into them throughout the semester. Here is a suggestion on how you might approach each assignment:
  - As soon as you receive an assignment, read it and understand what is being asked, even if you are not ready to commit any time to actually solving them. Ask questions if the specification is not clear. Once you have the problem in your head, it is most likely that you will find yourself thinking about it, perhaps at least subconsciously if not actively.
  - Ask questions when you get stuck. Please use Campuswire for asking questions. I am also available during office hours and the TAs are also able to help during their office hours. There is no need to schedule an appointment if you need help – just show up during the office hours.
  - *Start early so that you will have time to take a break when you get stuck.* Waiting until a few days before the due date is a bad plan. When I give you a week, I give you that much time because it would take that much, factoring in whatever else you would be doing in a typical week. It gets harder to get help from us as it gets close to the due date -- plus you won't have time to take a break by then.

- Think about your solutions. Your struggles on the assignment will be for naught if you don't review your work. Getting the right answer isn't enough. There is almost always a better way to solve a problem in programming.

Students with Disabilities

If you have a physical, psychological, medical or learning disability that may impact your course work, please let the instructor know. Reasonable accommodation will be provided if necessary and appropriate. All information and documentation are confidential.

Critical Incident Management

The University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Judicial Affairs any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn.