CSCI 1321 (Principles of Computer Science II), Spring 2024 - Hibbs: Syllabus

Course Description

This is the second course in computer science, following the guidelines established by the Association for Computing Machinery (ACM). It builds on the material from CSCI 1320, but begins to look seriously at object-oriented programming techniques as well as how to design and write proper code using that paradigm. It aims to improve your ability to solve problems using logical constructions that you plan, implement, and test. This course will further your ability to use more complex data structures and algorithms and help you to get a feel for how large programming projects can be developed and how you can make them more efficient, flexible, and sustainable. The course will additionally cover important technical capabilities foundational for future courses, including command line and version control infrastructure.

Learning Objectives

- **Collaboration.** Students will utilize collaborative problem solving and programming skills, such as pair programming, to approach CS problems.
- **Basic Object-Orientation.** Students will create and utilize classes with fields and methods in order to abstract and decompose larger problems, including through the use of access control, alternative constructors, and static members.
- **Inheritance Polymorphism.** Students will design and implement hierarchies of classes related through inheritance to abstract and generalize solutions to problems.
- **Parametric Polymorphism.** Students will create classes and functions applicable to multiple data types in order to abstract solutions and reduce code duplication.
- **Basic Data Structures.** Students will both implement and utilize Stacks, Queues, and List data types and will evaluate the computational complexity of these data structures.
- **Core Data Structure Usage.** Students will utilize additional core data structures, including Sets, Maps, Priority Queues, and Iterators, and will identify and solve problems suited using these structures.
- **Memory and References.** Students will write software demonstrating an understanding of how memory and variables are allocated, copied, modified, and shared, including by creating "deep" copies of data structures.
- **File Input/Output.** Students will create software that reads and writes from multiple files on the local filesystem.
- **Testing and Debugging.** Students will thoroughly test and debug their own software, including by utilizing exception handling and through formal unit testing and documentation.
- **Command line infrastructure.** Students will utilize command line interfaces to navigate through a file system, create/edit/delete files, and remotely interface with systems. Students will also compile and run their own programs from a command line interface.
- **Version control.** Students will utilize version control software (such as git) to develop software projects over time, and will utilize remote repositories to backup and submit assignments.

Basic information

Class meeting times and locations:

CSCI 1321-1:	MWF 12:30-1:20	CSI 257
CSCI 1321-2:	MWF 1:30-2:20	CSI 257

In general, sections of the course will be kept in sync, so if you need to occasionally attend the other section, you may do so. However, both sections of the class are full, so you may not have a seat.

Prerequisites: CSCI 1320 (Principles of Computer Science I) or Placement

Instructor contact information

Dr. Matthew Hibbs
Office: CSI 270K

- Office hours will be held regularly on most afternoons (except for Tuesdays):
  - Mon 3:30-5, Wed 2:30-5, Thurs 2-4, Fri 2:30-4*
    - *NOTE: Fri office hours will be canceled about once a month for faculty senate meetings
  - If something disrupts my regular office hour schedule, I'll do my best to update my public calendar, which is linked from the course website
  - Generally, if my door is open and I'm in my office, I'm available to meet with you. If I'm meeting with someone else in my office, please let me know that you're waiting for me so that I can let you know if you can come in or about how long it'll be for me to meet with you.
  - If my regular times don't work for you, please contact me by email and we can schedule a specific time to meet
- Email: mhibbs@trinity.edu
  - Email is generally a good way to contact me or ask questions. I try to be prompt, but I'm not always available, and may take more time to respond.
- Phone: (210) 999-7482

TA information

- Cole McGuire
- Office hours Tues 4-6 and Sun 5-7 in or around the CS department hallway (CSI 270)
- Email: cmcguir1@trinity.edu

Course materials

Textbooks

This course will use portions of several textbooks available freely for Trinity students through the Coates Library O'Reilly Book collection. (In case you haven't used this before, O'Reilly publishes many books related to programming and computer science, and your Trinity credentials give you access to their catalog of books.)

Portions of the following books will be assigned to cover the topics in this course:
Java 17 Quick Syntax Reference by Mikael Olsson (JQS in schedule)
Effective Java, 3rd Edition, by Joshua Bloch (EJ in schedule)
Object-Oriented Thought Process, 5th Edition, by Matt Weisfeld (referred to as OOTP in schedule)
Data Structures and Algorithms in Java, 2nd Edition, by Robert Lafore (DSAJ in schedule)

Web page

Most course-related information (this syllabus, homework and reading assignments, grades, etc.) will be made available via Google Classroom. The course web page is a starting point for online course material. Navigate to classroom.google.com and use the registration code provided on the first class day.

Other references

There are many great references on programming, algorithms, Java, git, and VSCode, far too many to list. The Internet is an amazing resource, especially for computer science, but as always, treat information from the web with caution. Some helpful resources are available on the links page of this website.

Course requirements

Grading

The grade for this course will be composed of five elements, discussed below. This table summarizes the contribution of each to your grade in the course. All items turned in for a grade in this course are pledged. For code, the pledge statement should be put in a comment at the top of the code.

Homework	25%
Tests (4 total; 12% each)	48%
Final Exam	20%
Class Participation	7%

Homework

Homework (primarily programming assignments) is a crucial part of this course; much of what you learn will likely be learned in the course of completing the programming assignments. Detailed requirements will be provided as part of each assignment; due dates will be announced via the course web page. The department's network of Linux machines contains all necessary tools for the course; but unless otherwise specified for individual assignments, you may use any system that provides a suitable environment.

For programming assignments and questions, ***only running code will be graded***. If an assignment submission contains syntax or other errors that prevent execution, you will receive no credit and your assignment will be returned with the option to re-submit with a point penalty. Depending upon the types of errors and the amount of work contained in the original answer, the re-submission penalty may range from 75% (no solution attempt made) to 10% (minor syntax errors) to 0% (typos of filenames) at my discretion. Only 1 round of re-submission per assignment is allowed. This is consistent with jobs and internships in the field: if you produce software that doesn't run, it isn't worth anything to a client or your employer.

A note on coding practices: In order to share and understand code (including helping me help you with your questions), basic coding practices are required, just as they are in the field. Uniform indentation and reasonable documentation are required. I may not help you to debug code that is not well-indented (well, I'll help, but I won't be happy about it...) Other coding choices (such as bracket alignments or amount of whitespace) are at your discretion, but you should be uniform, and all blocks of code should be indented beyond the surrounding code.

An Important Note: MANY HOMEWORKS REQUIRE SPECIFIC FILE/ PACKAGE/ CLASS/ FIELD/ METHOD NAMES. ***THESE ARE IMPORTANT!*** I HAVE SCRIPTS THAT I USE TO HELP GRADE ASSIGNMENTS THAT ARE DEPENDENT ON YOU FOLLOWING THESE CONVENTIONS. HOMEWORK THAT DOES NOT CONFORM TO NAMING REQUIREMENTS WILL BE "BOUNCED" BACK FOR CORRECTION BEFORE GRADING. I WILL NOT GRADE HOMEWORK WITH INCORRECT FILE/PACKAGE NAMES.

Tests

There will be 4 tests in class throughout the semester focused on the most recently covered material. These tests will be taken in class and will require that you use the classroom computers to complete them. Tests will include a combination of coding and short answer questions, and will be administered using a platform where you can run your code before submitting. Unlike homework assignments, code that does not run may still receive partial credit.

Although subject to change depending on how quickly we cover material, the planned dates of tests are:

- Friday 2/9
- Friday 3/1
- Wednesday 3/27
- Monday 4/22

Please refer to the course schedule page for the most up-to-date schedule.

Final Exam

The Final Exam will be comprehensive of all material covered throughout the semester. The format will be similar to the in class tests, but will be more lengthy. The final exam will be administered using the same platform as the in class tests, and partial credit may be awarded to code that does not run.

- Final Exam: Tues 5/7 @ noon (12:30 section)
  Tues 5/7 @ 3:30pm (1:30 section)

Class participation

Regular class attendance is expected; class participation grades will be based on attendance, engagement, and participation in class discussions and activities. Attendance will be taken through "participation polls" online at the end of each class.

Course Policies

Classroom environment

I like my classes to be interactive. I ask lots of questions, and I like to get your answers. There will also be occasional small group activities completed during class. These help me understand how well you are understanding the course material and whether I need to speed up or slow down through given material. As detailed above, a portion of your grade is based on class participation, which includes answering questions, participating in discussions, and working on activities. No texting/email/videos/League/facebook/etc. during class. You can't participate if you're not paying attention.

Late and missed work

Exams can be made up only in cases of documented conflict with a university-sponsored activity, documented medical emergency, conflict with a religious holiday, or with PRIOR APPROVAL from me. Unless otherwise stated for a particular assignment, homework will only be accepted late with permission from me. Contact me by e-mail or in person PRIOR to the due date if you require an extension. If an extension is granted, it may carry a grade penalty depending on the circumstances and length of extension. Only 1 extension may be granted per assignment.

Academic integrity at Trinity

All students are covered by a policy that prohibits dishonesty in academic work. Under the Honor Code, a faculty member will (or a student may) report an alleged violation to the Academic Honor Council. It is the task of the Council to investigate, adjudicate, and assign a punishment within certain guidelines if a violation has been verified. **Students are required to pledge all written work that is submitted for a grade: “On my honor, I have neither given nor received any unauthorized assistance on this work” and their signature. The pledge may be abbreviated “pledged” with a signature.**

Collaboration and academic integrity in this course

All work submitted for a grade (homework assignments, quizzes, and exams) must represent the student's own individual effort. All graded work will be considered pledged work.

Discussion of homework assignments among students is allowed (and encouraged!), but not to the point where specific answers or code are being written collectively. In short, don’t share answers, either in person or electronically. All assignments are subjected to comparison with other solutions, including solutions from prior semesters. Code that is deemed unusually similar to other work will be subject to disciplinary actions.

Generative AI can be a powerful and helpful tool for computer science; however, it is vital to innately understand the fundamentals of CS and programming in order to responsibly use these tools. As such, **the use of generative AI tools (such as ChatGPT, Co-pilot, Ghostwriter, etc) is prohibited**.

Some examples of ***acceptable*** collaboration are:

- Discussing a general approach to a problem with a fellow student, such as deciding whether loops or recursion are best to write a program.
- Asking an ACM tutor for help with specific syntax questions.
- Comparing program outputs with fellow students to decide if you are getting the correct results.
- *After the due date of an assignment* (and after you've turned it in), comparing your code with fellow students to discuss alternative solutions.

Some examples of ***unacceptable*** collaboration are:

- Pasting a portion of your code into an email to anyone other than me.
- Writing exactly (or nearly exactly) the same code as a fellow student while sitting side-by-side.
- Utilizing an approach given to you by a tutor or fellow student without understanding the rationale or logic behind that approach.
- Using an AI tool, such as ChatGPT or Co-pilot, to draft an initial codebase, either in whole or in part.
- *Before you've turned in an assignment*, comparing your code with fellow students to discuss alternative solutions.

If you are uncertain about whether a particular level of collaboration is acceptable, please ask for clarification. Please also note when you turn in an assignment whether you sought help with it from other students or faculty (e.g., "J. Random and I worked on this assignment together" or "I got help with this assignment from one of the ACM tutors"). Answers that are identical beyond coincidence (either to another student's work or to a sample solution from a previous semester) will be considered to be in violation of the Honor Code, and will result in appropriate action. You are responsible for the security of your work, both electronic and hard copy.

Title IX/Sexual Misconduct Reporting

As a Responsible Employee who is committed to creating an environment where every member of our community can thrive, I want to let you know that I am a Mandatory Reporter under Texas state law. What that means is that I am am required to report any instances of sexual misconduct, including sexual harassment, non-consensual sexual intercourse, non-consensual sexual contact, sexual exploitation, intimate partner violence, stalking, and related retaliation that I am aware of to the Title IX Coordinator. So, if you share information with me about any incidents that implicate the Sexual Misconduct or Anti-Harassment Policies, I am required to report all information to the Title IX Coordinator to make sure you have information about support resources and complaint resolution options. My report does not initiate the complaint process, and you are in control over how you choose to engage with our Title IX Coordinator. If you or someone you know has experienced sexual misconduct, including sexual harassment, I encourage you to share this information directly with the Title IX Coordinator or one of the individuals who has been designated as a confidential resource on campus. Information about reporting is available here: Reporting.

Academic Support Resources

Trinity faculty hold students to the highest academic standards and also know that the very best students seek out help when necessary. The following resources are in place to support your academic success:

- Academic Success: time management, student skills, test anxiety, note taking, supplemental 1:1 tutoring
- Career Services: major exploration, career guidance
- Counseling Services: mental health concerns, mental health referrals
- Quantitative Reasoning and Skills Center: tutoring for quantitatively demanding coursework
- Student Accessibility Services: accommodations for a diagnosed disability
- Wellness Center: nutrition, sleep, stress management
- Writing Center: starting a paper, finding a thesis, drafting and editing