COMP 142 — Computer Science II: Objected-Oriented Programming — Fall 2021
CRNs 12387/12388

Instructor:	Phillip Kirlin
Times:	(Monday 12–1:50 and WF 12–12:50) or (Monday 2–3:50 and WF 1–1:50)
Classroom:	Briggs 001
Course website:	cs.rhodes.edu/142
Email:	kirlinp@rhodes.edu (please include "CS 142" somewhere in the subject)
Office:	Briggs 209
Office hours:	See website for scheduled office hours. I am also available by appointment.

Official Course Description: An introduction to the fundamental concepts and practices of object-oriented programming. The object-oriented programming paradigm is introduced, with a focus on the definition and use of classes as a basis for fundamental object-oriented program design. Other topics include an overview of programming language principles, simple analysis of algorithms, basic searching and sorting techniques, and an introduction to software engineering issues.

Unofficial Course Description: CS142 is the second course in the sequence for computer science majors or minors and ideally should be taken immediately after CS141. CS142 offers a new perspective on software design through an introduction of the object-oriented paradigm. Special emphasis is placed on the process of building hierarchies of abstractions to hide implementation details through a careful and systematic analysis of problems of moderate complexity. Various design approaches will be explored with the goal of identifying the situations for which each approach is applicable.

This course will use the Java programming language as the vehicle for exploration of fundamental computer science concepts. However, this is not a course about Java; it is about the structure and interpretation of computer programs.

The particular Java environment that will be used in this course is available in the computer labs on Rhodes College campus, and can also be installed on personal computers. You are free to develop the code for the assignments on your own computer using an environment of your choice. However, keep in mind that the source code that you submit for assignments must compile and run successfully using the environment from class.

Course Objectives: At the end of this course, you should be able to

- analyze problems of moderate complexity and solve such problems by writing code using the Java programming language,
- apply principles of good program design, especially the uses of data abstraction and modular program composition,
- understand the fundamental design, analysis, and implementation of basic data structures and algorithms,
- assess how the choice of data structures and algorithm design methods impacts the performance of programs,
- apply your programming skills to diverse applications in science and engineering

Course Topics: (not necessarily in this order)

- Java basics: syntax, data types, arrays, strings, functions, file reading
- Objects and classes
- Object-oriented design
- Inheritance and polymorphism
- Recursion
- Asymptotic analysis
- Linked lists
- Optional topics as time permits: generics, other data structures (stacks, queues)

Text: There is no required textbook for this class. Materials will be distributed in class or online.

Prerequisites: The course assumes successful completion of CS141 or significant programming experience. Please come see me if you have not had CS141.

Coursework:

	Tentative weight	Tentative date
Programming projects	40%	
Labs and homework	15%	
Midterm 1	12.5%	Wednesday, September 29, in class
Midterm 2	12.5%	Wednesday, November 10, in class
Comprehensive final exam	20%	Wed, Dec 15, 1pm (12pm WF section) Sat, Dec 11, 1pm (1pm WF section)

Grades of A–, B–, C–, and D– are guaranteed with final course grades of 90%, 80%, 70%, and 60%, respectively. If your final course grade falls near a letter grade boundary, I may take into account participation, attendance, and/or improvement during the semester.

Assignments are due at 11:59pm unless otherwise specified. In general, late will work will not be accepted without an extension obtained prior to the due date.

Office Hours: In addition to regular office hours, am also available immediately after class for short questions. You never need an appointment to see me during regular office hours; you can just come by. Outside of regular office hours, feel free to stop by my office, and if I have time, I'll try to help you. If I don't have time at that moment, we'll set up an appointment for a different time. Don't be shy about coming by my office or sending me email if you can't make my regular office hours. I always set aside time each week for "unscheduled" office hours.

Attendance: Attendance is expected for each class. If your attendance deteriorates, you will be referred to the dean and asked to drop the course. Attendance, participation, and apparent overall improvement trend may be considered in assigning a final grade. Attendance will be checked each class lecture period. After five unexcused absences, each additional absence will reduce the final grade for the course by one letter grade.

Workload: It is important to stay current with the material. You should be prepared to devote at least 2–3 hours outside of class for each in-class lecture. In particular, you should expect to spend a significant amount of time for this course working on a computer trying example programs and developing programming assignments. Do not wait to the last minute to start your programming assignments.

You are encouraged to form study groups with colleagues from the class. The goal of these groups is to clarify and solidify your understanding of the concepts presented in class, and to provide for a richer and more engaging learning experience. However, you are expected to turn in your own code that represents the results of your own effort.

Programming Assignments:

- All programs assigned in this course must be written in Java, unless otherwise specified. When turning in assignments, submit only the Java source code files (`.java`); do not submit any files generated by the IDE (e.g., `.class`).
- Back up your code somewhere as you're working on your assignments. Computer crashes or internet downtime are not valid excuses for missing a deadline.
- Programming grades will be graded on correctness of the program output, efficiency and appropriateness of the algorithms used in the code, and style and documentation of the source code.
- Grades are assigned to programs as follows by this general guideline:
  - A (100 pts): The program is carefully designed, efficiently implemented, well documented, and produces clearly formatted, correct output.
  - A- (93 pts): The program is an 'A' program with one or two of the minor problems described for grade 'B.'
  - B (85 pts): The program typically could easily have been an 'A' program, but it may have minor/careless problems such as poor, inadequate, or incomplete documentation; several literal values where symbolic constants would have been appropriate; wrong file names (these will be specified per program assignment); sloppy code format; minor efficiency problems; etc. (This is not an exhaustive list.) You would be wise to consider 'B' the default grade for a working program — this might encourage you to review and polish your first working draft of an assignment to produce a more professional quality final version of your program.
  - C (75 pts): The program has more serious problems: incorrect output or crashes for important special cases (the "empty" case, the "maxed-out" case, etc.), failure to carefully follow design and implementation requirements spelled out in the assignment, very poor or inefficient design or implementation, near complete absence of documentation, etc.
  - D (60 pts): The program runs, but it produces clearly incorrect output or crashes for typical cases. Or, it may deviate greatly from the design or implementation requirements stated in the assignment description.
  - F (35 pts): Typically, an 'F' program produces no correct output, or it may not even run. It may "look like a program" when printed as a hard copy, but there remains much work to be done for it to be a correct, working program.

Rules for Completing Assignments Independently

- Unless otherwise specified, programming assignments handed in for this course are to be done *independently.*
- Talking to people (faculty, other students in the course, others with programming experience) is one of the best ways to learn. I am always willing to answer your questions or

provide hints if you are stuck. But when you ask other people for help, sometimes it is difficult to know what constitutes legitimate assistance and what does not. In general, follow these rules:

- **Rule 1: Do not look at anyone else's code for the same project, or a different project that solves a similar or identical problem.**
  Details: "Anyone else" here refers to other members of the class, people who have taken the class before, people at other schools enrolled in similar classes, or any code you find online or in print. "Similar or identical problem" here should allow you to look at code that uses techniques applied in different situations that you can then adapt to your project. However, if you find yourself copying-and-pasting code or directly transforming code line by line to fit into your program, then that is considered plagiarism.
  Exception: You may help someone else debug their program, or seek assistance in debugging yours. However, this requires the person writing the code being debugged to have made a good-faith attempt to write the program in the first place, and the goal of the debugging must be to fix one specific problem with the code, not re-write something from scratch.
- **Rule 2: Do not write code or pseudocode with anyone else.**
  Details: You must make a good faith effort to develop and implement your ideas independently before seeking assistance. Feel free to discuss the project *in general* with anyone else before you begin and as you're developing your program, but when you get to the level of writing code or pseudocode, you should be working independently.

The underlying idea is that you are entitled to seek assistance in ways which will genuinely help you to learn the material (as opposed to just getting the assignment done). Programming assignments are graded as a benefit to you; they are your chance to show what you have learned under circumstances less stressful than an exam. In return, I ask only that your work fairly reflect your understanding and your effort in the course.

Coding Style: Designing algorithms and writing the corresponding code is not a dry, mechanical process, but an art form. Well-written code has an aesthetic appeal while poor form can make other programmers (and instructors) cringe. Programming assignments will be graded based on correctness and style. To receive full credit for graded programs, you must adhere to good programming practices. Therefore, your assignment must contain the following:

- A comment at the top of the program that includes the author of the program, the date or dates, and a brief description of what the program does
- Concise comments that summarize major sections of your code, along with a comment for each function in your code that describes what the function does.
- Meaningful variable and function names
- Well-organized code
- White space or comments to improve legibility
- Avoidance of large blocks of copy-and-pasted code

Class Conduct:

- I encourage everyone to participate in class. Raise your hand if you have a question or comment. Please don't be shy about this; if you are confused about something, it is likely that someone else is confused as well. Teaching and learning is a partnership between the instructor and the students, and asking questions not only helps you understand the material, it also helps me know what I'm doing right or wrong.
- Do not use your cell phone while in class, and keep the ringer on silent.
- If you cannot make it to class for whatever reason, make sure that you know what happened during the lecture that you missed. It is your responsibility, and nobody else's, to do so. The best way to do this is to ask a classmate.
- If you have to leave a class early, inform the instructor in advance. It is rude to walk out in the middle of a lecture.

Access and Accommodations: Your experience in this class is important to me. If you anticipate or experience physical or academic barriers based on disability, please let me know immediately so we can discuss options. If you have already established accommodations with Student Accessibility Services (SAS), please communicate your approved accommodations to me at your earliest convenience so we can discuss your needs in this course.

If you have not yet established services through SAS, but have a condition that requires accommodations (conditions include but not limited to mental health, attention-related, learning, vision, hearing, physical or chronic health), please contact SAS at 901-843-3885, Burrow Hall 4th floor, or by visiting `www.rhodes.edu/accessibility`.

Academic Integrity: Plagiarism, cheating, and similar anti-intellectual behavior are serious violations of academic ethics and will be correspondingly penalized. If you are concerned about a possible violation of this kind, please talk with me. I understand that being a student at Rhodes can be stressful sometimes and you will have many demands on your time. However, I would much rather have you turn in a partially-completed assignment or do poorly on a test than have you violate the Rhodes Honor Code. I can — and very much want to — help you if you don't understand the material, but violations of academic integrity will be dealt with harshly.

Unless otherwise specified, everything you submit in this course must be your own work and represent your individual effort. These are all included in the definition of reportable Honor Code violations for this course: copying all or part of a solution to a problem, downloading a solution from the internet and submitting it as your own, having someone else provide the solution for you, or allowing someone else to copy from you. If you have any doubt about what type of behavior is acceptable, please talk with me.

Diversity: A diverse learning community is a necessary element of a liberal arts education, for self-understanding is dependent upon the understanding of others. We are committed to fostering a community in which diversity is valued and welcomed. To that end any discrimination or harassment on the basis of race, gender, color, age, religion, disability, sexual orientation, gender identity or expression, genetic information, and national or ethnic origin, will not be tolerated in the classroom.

We are committed to providing an open learning environment. Freedom of thought, a civil exchange of ideas, and an appreciation of diverse perspectives are fundamental characteristics of a community that is committed to critical inquiry. To promote such an academic and social environment we expect integrity and honesty in our relationships with each other and openness to learning about and experiencing cultural diversity. We believe that these qualities are crucial to fostering social and intellectual maturity and personal growth.

Intellectual maturity also requires individual struggle with unfamiliar ideas. We recognize that our views and convictions will be challenged, and we expect this challenge to take place in a climate of open-mindedness and mutual respect.

Sexual Misconduct Disclosure: Rhodes is committed to ensuring a safe learning environment that supports the dignity of all members of the Rhodes community. Rhodes prohibits and will not tolerate sexual misconduct, which includes, but is not limited to, dating/domestic violence, sexual assault, sexual exploitation, stalking, sexual harassment and sex/gender discrimination. Rhodes strongly encourages members of the Rhodes community to report instances of sexual misconduct immediately. All Rhodes faculty, staff, Peer Advocates, and Resident Assistants are Mandatory Reporters (exceptions are confidential resources: Counseling Center, Chaplain, and Student Health Center) and are required by the College to report any knowledge they receive of possible violations of this policy to the Title IX Coordinator. If you choose to share information related to sexual misconduct with me, I will report it to the Title IX Coordinator; however, you will control how your report is handled and you are not required to pursue a formal claim. The goal is to make you aware of the range of options and resources that are available to you. For more information about Rhodes' sexual misconduct policy or to make a report please see `www.rhodes.edu/titleix`.