CSC121 - Object-Oriented Program Design - Fall 2024

Lecture:

MWF 9:00 AM-9:50 AM, McAllister Hall 233

Instructor:

Nadeem Abdul Hamid

Contact:

nhamid@berry.edu • @nadeemabdulhamid • 706.368.5632

Student Hours:

Monday/Wednesday: 10am-1pm

Tuesday/Wednesday: 2-3pm

Thursday: 9am-11am

or by appointment.

Course Description

(3-0-3) An introduction to the object-oriented model of programming, building on the design methodology introduced in CSC120. Students will design, implement, test, and debug object-oriented programs composed of multiple classes. Topics include classes, objects, inheritance, exceptions, interfaces, design by contract, basic design patterns, and software reuse. PR: CSC120.

Textbooks

- M. Felleisen, M. Flatt, R.B. Findler, K.E. Gray, S. Krishnamurthy, V.K. Proulx, *How to Design Classes*. (Optional reading to follow along with lecture topics, first 7-8 weeks.)

Technology

- Canvas course site: http://cs.berry.edu/csc121.

> It is your responsibility to check the web site for this course regularly (i.e. daily) throughout the semester, as it will be regularly updated with announcements, lecture notes, source code, assignments, etc.

- Java OpenJDK (install first) and Eclipse IDE
- Github Desktop

Purpose of the Course

Picking up from CSC120, the purpose of this course is to introduce you to an object-oriented model of programming and computation. The goal is to help you understand the principles of class-based program design using an object-oriented programming language(s) - *i.e.* the goal is not Java itself. We use Java so you can see how the principles we learn are used in practical application. With emphasis on the design process, the course studies class-based program design and the design of abstractions that support reusable software and libraries.

Student Learning Outcomes

By the end of the course, you should be able to:

- Understand how to design programs using the object-oriented paradigm
- Demonstrate the ability to implement these designs
- Produce well-documented implementations
- Gain exposure to state-of-the-art development tools
- Use critical thinking skills and creativity to solve problems

Assessment Measures

The determination of how well a student meets the learning outcomes above will be assessed by their overall grade in the course, based on performance on assignments and exams, as detailed below (see 'Evaluation Components'). A satisfactory level of competency will be demonstrated by earning a grade of 75% (C) or higher in the course.

Evaluation Components and Grading Scale

Your grade will be computed based on a weighted average of the following course requirements. The minimum to attain an A will be an overall course average of 90%, for a B 80%, for a C 70%, and for a D 60%.

- **10% Class attendance/participation**

  > Attendance in class is essential. Our meetings will consist of a mixture of lecture, discussion, and programming. You will be expected to participate actively in class.

- **50% Programming assignments**

  > Assignments will usually be assigned as homework at the end of each class meeting and are intended to be completed for evaluation by the next class meeting period.. These assignments will involve applying techniques learned in class and will often involve extending or modifying code originally developed in a lecture session. Each programming homework task will be graded as "Complete" or "Incomplete" based on the specifications provided with the assignment. *Incomplete* tasks may be reworked, with the instructor’s assistance, until they are *Complete*.

- **10% Semester project**

  > You will work in groups of 2 or 3 people on a project throughout the semester. Details of the project will be provided as the course progresses. It will involve the design and development of a interactive, graphical computer game or simulation chosen by each group.

- **10% Quizzes**

  > Online quizzes will be posted approximately every week. They will cover material from lecture and homework. These will be done individually and outside of class, but will be time-limited. You may use the textbook, lecture notes from the course web page, source code from class and assignments, and

> any notes that you have taken yourself for these, but no Internet resources (unless explicitly noted otherwise).

- **20% Exams**

> There will be a midterm (Friday, October 11th) and a final exam (Monday, December 9th, 11am-1pm)

Pair Programming

You may work with a partner for the programming assignments if you like. If you do, you **must** engage in the pair programming model, that is, both of you should be sitting at a single machine, side by side. One of you is "driving," at the keyboard and mouse programming. The other is actively engaged "navigating" - following along, catching mistakes, and providing ideas. You should make sure that over the course of an assignment that you spend roughly the same amount of time each "driving" and "navigating". When you submit your work, make sure **both names** are on **every** file.

Note: the following are not pair programming practices and are **not acceptable** ways of working with a partner for this course.

- Splitting the work in half, where each person does part of the work.
- Working individually at separate times.
- One person doing all of the work, but turning it in with both partner's names.

These will most likely reflect in poor exam grades since one or both partners will not be as familiar with the entire material as they should be.

Important note 1: A "pair" is two people. You are *not* allowed to work with more than one other person on programming assignments. Any assignment you turn in with more than 2 names on it will be returned with a zero (0) grade.

Important note 2: DO NOT share your (pair's) code with others. If you start working on homework exercises with one partner, do not switch halfway through to another partner, and share the code that you already completed.

Academic Integrity

Students are expected to have read carefully and understood the rules governing breaches of academic integrity that are to be found in the Viking Code and the Course Catalog. Incidents of cheating will be dealt with as detailed in the Viking Code and Course Catalog. For the purposes of this course, we adopt the following[^1] policies and philosophy:

> This course's philosophy on academic honesty is best stated as "be reasonable." The course recognizes that interactions with classmates and others can facilitate mastery of the course's material. However, there remains a line between enlisting the help of another and submitting the work of another. The course's policy characterizes both sides of that line.
>
> ***The essence of all work that you submit to this course must be your own.*** Unless otherwise specified, collaboration on evaluated activities (e.g., assignments, labs, problem sets, projects, quizzes, or tests) is not

> permitted except (1) you are allowed work in pairs as described above if you wish to do so, and (2) to the extent that you may ask classmates and others for help so long as that help does not reduce to another doing your work for you. Collaboration on assignments may be permitted to a greater extent if explicitly specified in the assignment.
>
> Generally speaking, feel free to discuss homework questions with others in general terms, but you must not take any written or electronic notes out of these discussions. Cheating in any form - including copying someone else's work **or letting your work be copied** - is unacceptable. Please don't do it. When asking for help, you may **describe** your issue(s) to others in general terms, but you may not show them substantial portions of your code or view theirs, so long as you and they respect this policy's other constraints. Collaboration on quizzes or exams is not permitted at all.
>
> Below are rules of thumb that (inexhaustively) characterize acts that the course considers **reasonable** and **not reasonable**. If in doubt as to whether some act is reasonable, do not commit it until you solicit and receive approval in writing from the instructor. Acts considered not reasonable will be subject to **harsh and strict penalty, up to and including an failing grade (F) in the course or administrative withdrawal**. Violations of this policy will be dealt with as detailed in the Viking Code and Course Catalog and **will be reported to the Provost**, as required by Berry College policy.
>
> *[Escape Clause]* If you commit some act that is not reasonable but, on your own initiative, bring it to the attention of the instructor within 72 hours of its occurrence, localized sanctions may be imposed, such as a partial, incomplete, or failing grade for work submitted, but the matter will not be referred or reported for further disciplinary action except in cases of repeated acts.

If you are at all unsure about the appropriateness of a particular course of action, please message me on Discord or email and ask for help, to be safe.

Reasonable

- Communicating with classmates about assignment tasks in English (or some other spoken language), without recording anything in written or electronic form for the duration, and properly citing those discussions.
- Discussing the course's material (including anything in the textbook and/or lecture slides/videos) with others in order to understand it better.
- Helping a classmate identify a specific bug in his or her code at office hours, elsewhere, or even online, as by viewing a few lines (< 10) of his or her code, and only offering verbal suggestions of potential fixes. (No hands on keyboard; no sharing your screen or files.)
- Incorporating a few lines (< 10) of code that you find online or elsewhere into your own code, provided that those lines are not themselves solutions to assigned problems and that you ***cite*** the lines' origins.
- Turning to the web or elsewhere for instruction beyond the course's own, for references, and for solutions to technical difficulties, but not for outright solutions to homework problems or the semester project.
- Using Internet resources to search for general help and information about basic coding techniques or technical difficulties (for example, searching Google or Stack Overflow for "how to declare an ArrayList in Java", or "out of bounds error message Java"), but not to search specifically for techniques related

specifically to the topics of the course or for outright solutions to an assignment (for example, do not search for "how to implement a linked list in Java").
- Whiteboarding solutions to assignments with others using diagrams or pseudocode (English outline) but not actual code.
- Working with (and even paying) a tutor to go over course material and examples with you, provided the tutor does not do your work for you.
- You are always welcome to consult the instructor for assistance - in person, by email, Discord, etc. - if you are stuck.

Not Reasonable

- Accessing a solution to an assignment (e.g. looking at a code file on someone else's computer, or downloading one to your own) prior to (re-)submitting your own.
- Asking a classmate to see their solution to an assignment before (re-)submitting your own.
- Giving or showing to a classmate a solution to an assignment when it is they, and not you, who is struggling to solve it.
- Decompiling, deobfuscating, or disassembling instructor-provided solutions to assignments.
- Failing to cite (as with comments) the origins of code or techniques that you discover outside of the course's own lessons and integrate into your own work, even while respecting this policy's other constraints.
- Looking at another individual's work during a test or quiz.
- Paying or offering to pay an individual for work that you may submit as (part of) your own, or having them "walk you through" an assignment, describe in detail how to solve it, or sit with you as you write a solution/answer.
- Providing or making available solutions to assignments to individuals who might take this course in the future.
- Searching for or soliciting outright solutions to assignments online or elsewhere.
- Splitting an assignment's workload with another individual and combining your work (unless specifically permitted in the assignment instructions). If you work on an assignment with a partner, please ensure **both your names** are clearly recorded in a comment **at the header of all your source code files**.
- Submitting (after possibly modifying) the work of another individual beyond the few lines allowed by this policy.
- Submitting the same or similar work to this course that you have submitted or will submit to another.
- Viewing another's solution to an assessment and basing your own solution on it.
- Using third-party generative AI-based software (including ChatGPT, GitHub Copilot, the new Bing, et al.) that suggests answers or lines of code.

[^1]: Adapted from Harvard University's CS50 Syllabus, "Academic Honesty": https://cs50.harvard.edu/x/2023/honesty/.

As a general rule, if you do not understand what you are handing in, something is probably wrong. If you have given somebody some code simply so that it can be used in that person's assignment, you are probably cheating. Please don't do it.

The use of Internet q&a sites or generative AI tools to complete assignments is prohibited for this course. These have the potential to be useful for experienced developers and/or niche tasks. However, just as using a life vest (PFD) while learning to swim is *counterproductive, detrimental, and even dangerous*, relying on such tools will

impede and prolong the process of your learning the material that this course covers. As noted in this Cornell University report, "In an introductory programming course, ... GAI code generation is ***actively harmful to student learning***."

> If anyone is found to be using ChatGPT or similar tools to generate code solutions for assignments in this course, without express permission or direction to do so, they will be withdrawn administratively from the course.

Attendance Policy

Please see the Berry College Viking Code for "Class Attendance Policies". Missing three (3) or more classes without justifiable reason (and appropriate documentation) will be considered excessive absences, an alert form will be sent to the proper authorities, and will be grounds for administrative withdrawal from the course at the instructor's discretion.

Accommodation Statement

The Academic Success Center provides accessibility resources, including academic accommodations, to students with diagnosed differences and/or disabilities. If you need accommodations for this or other classes, please visit http://berry.edu/asc for information and resources. You may also reach out at 706-233-4080. Please note, faculty are not required, as part of any temporary or long-term accommodation, to distribute recordings of class sessions.

Electronic Devices

Please don't use cell phones, pagers, etc. in class. Do not use computers for other than course activities during class time. Texting, instant messaging, emailing, social media'ing, and similar activities during class are inappropriate and distracting to everyone. Be respectful.

Schedule of Class Sessions

A detailed schedule for the semester (subject to change) is available separately.

Acknowledgements

The ideas and material for this course are drawn and built on substantial amount of work and content prepared by Kathi Fisler and others for CS 2102 (Object-Oriented Design Concepts) at Worcester Polytechnic Institute; Viera Proulx, Benjamin Lerner, and others for CS 2510 (Fundamentals of Computer Science 2) at Northeastern University; and Jennifer Walter, Vasser College.

Note: All information, schedules, and policies in this document are subject to change at the instructor's discretion. The latest version of this syllabus will always be available on the online course website.