

# CPSC 240 — Object-oriented Analysis & Design

Professor: Stephen Davies  
Spring 2024

**Lecture:**

MF 11–11:50am, W 11am–12:50pm, Farmer 054  
...or...  
MF 2–2:50pm, W 2–3:50pm, Farmer 054

**Office hours (Farmer 044):**

Mondays 12–2pm  
Tuesdays 3:30–4:30pm  
Wednesdays 10–11am  
Thursdays 1–2pm

**Final exam:** Fri May 3rd, Noon-2:30pm ...or... 3:30-6pm

<http://stephendavies.org/cpsc240>

---

## Design vs. code

The precursor to this course (CPSC 220) had the word “programming” in its title. This one, in contrast, features the words “analysis” and “design.” Is this just semantics, or is something deeper implied by this choice of terms?

The wording was deliberate, and in many ways it’s a matter of *scale*. When we talk about **programming**, we think of individual lines of code that are assembled to create a function or a loop. But the word **design** connotes something larger: a system complex enough that it requires advanced planning and organization to keep it under control. Programming is tactical: your focus is on what’s in front of you and on getting the details right. Design is strategic: your focus is on the big picture and on outlining a structure into which all those details can fit together.

Although at this point of your career you’ve acquired some essential programming expertise, two important things have probably been missing from your experience: (1) big projects, and (2) team projects. The early courses in any computer science curriculum focus on tiny programs that you complete on your own. They may not have seemed tiny at the time, but if you compare them with anything you use on a

daily basis – Excel, Firefox, YouTube, Fortnite – you’ll realize just how small they really are.

**It turns out that once a software project reaches a particular size, it undergoes certain changes that make it unlike anything smaller.** That’s not only because of increasing complexity, but also because it’s now too big for any one person to hold in their head at once. You now have to depend on an intuitive design and accurate documentation to have a fighting chance of writing the code correctly. This is an utterly new kind of experience.

Working cooperatively with others is also a new kind of experience. You have to depend on someone else’s code working right in order for yours to work right. This is nearly always how software development works in the real world, and gaining practice with this paradigm will be a primary focus of this class.

When these two pieces of the puzzle are in place, you will have acquired an important level of mastery. By the end of this course, you will know enough about how software is really built to be professionally hire-able. You should be able to walk in to your average development team as a ground-floor software engineer and basically know what the heck you’re doing. It will be challenging, but also worth it in the long run. So hang on and enjoy the ride.

## Course objectives

To provide students with:

- opportunities to design, implement, debug, and document original larger-scale programs using techniques of good object-oriented design
- experience using design techniques for identifying classes and methods
- an introduction to UML including class diagrams and sequence diagrams
- opportunities to study the role of encapsulation, abstraction, visibility, inheritance and polymorphism in object-oriented design
- a collaborative, team development experience
- an introduction to the concept of design patterns and the application of common patterns

- feedback to help students improve their writing skills needed in computer science
- a small taste of multi-threaded (parallel) programming concepts, in preparation for further development of these ideas in later courses

## Student Learning Outcomes

Upon completing this course, students will be able to:

- recognize and explain the role of encapsulation, visibility, inheritance and polymorphism in object-oriented design
- identify and document requirements using Use Cases
- use design techniques to identify classes and methods and capture them in UML class diagrams
- identify test cases and apply them to a project
- develop larger scale programs in a collaborative, team environment
- recognize design patterns and the application of common patterns
- recognize diversity and inclusion issues and describe their effects within the field

## Rules of the game

1. There are NO stupid questions. I will never belittle you or make you feel dumb about anything. Your job is not to already know everything before you start, but to roll up your sleeves and work hard to try and learn. Even if your question is, “Stephen, I totally didn’t even get that, can you start over and explain it again?”, please ask.
2. This class will be interactive. When I point at you in class, say your first name, and be prepared to try and answer questions. (Don’t worry if you don’t know “the answer” every time. Go ahead and give it a shot, or else just say “**pass**” which is always accepted.)

3. Except where explicitly indicated, you must complete all work for this class entirely on your own, without any help from any other source (see “The Honor Code and this course,” below).
4. Exams and quizzes will be based on both the book readings and the material I cover in class. Make sure you know both!
5. Please, **no laptops during lecture**. I’ve had students claim that they take notes on their laptop during class, but even if it’s true, those things are too big a distraction to you and your fellow students to make it worth it. Just stay tuned in, because I move fast.

## Books

We’ll be using my book *Blueprints: Creating, Describing, and Implementing Designs for Larger-scale Software Projects* this semester. (This is an *open* textbook, which means among other things that I get **no royalties** from it.) You can get a copy from the UMW Bookstore, or from Blurb.com (cheaper but takes longer to arrive) or Amazon (more expensive but arrives quicker).

## Calendar

The calendar for the course, complete with assignment due dates, tests, *etc.*, will be maintained on the course website <http://stephendavies.org/cpsc240>. The “Road Map” at the end of this syllabus is a general guide to the order we’ll be covering things, but the definite present is always located on the website.

## Late policy

No late work will be accepted this semester. Get your stuff in on time, there’s no excuse not to!

## Grading

Activity	% of grade
Six quizzes	20%
Six team SWYK (“Show What Y’all Know”) checks	20%
Individual programming and non-programming assignments	20%
Team design and programming assignments	20%
Final exam	20%

### Writing assignments

Although not a “Writing Intensive” course *per se*, this class will indeed involve writing some English text. Sometimes this will be a short responsive essay on a Computer Science topic; other times a technical artifact such as Use Case descriptions or API documentation.

The thing I want to emphasize right away is that in our discipline **writing well does actually matter and is something you should actually care about**. This comes as a surprise to many CPSC students. But our discipline is full of ideas, and ideas must often be expressed in writing. Get good.

### Grading team assignments

For the SWYK checks, you’ll be assigned **a random partner** at the start of Friday’s lecture. You and your partner will be given a handout describing what your code needs to do, and the two of you will have one hour of “pair programming” in which to complete it. You’ll both get the same grade for the SWYK check, and you’ll never have the same partner more than once in the semester.

For the team project, you will be assigned to **teams of three** the last week of February. (The designation of students to teams is at my discretion, but if you feel strongly about someone you do not want to work with, you may mention this preference to me.) All team activities will be performed as a team, and **grades will be awarded (pretty much) uniformly to all team members**. I reserve the right to tweak individual grades up or down on the team project based on what I observe, but this will only be a tweak.

## The Honor Code and this course

I strongly believe in UMW's honor code and scrupulously adhere to it. Here are the rules for this course:

1. The quizzes and final exam are to be solely your own work. They are open book and open notes, but they are both **closed Java compiler** and **closed to other humans**.
2. For the *individual* homework assignments, you must work **entirely on your own**. To be clear:
  - Discussing an assignment with a fellow student in the course is a no-no.
  - Discussing an assignment with a previous student from this course is a no-no.
  - Discussing an assignment with anyone who has never even taken the course is a no-no.
  - Googling for code snippets or design ideas is a no-no.
  - Referencing StackOverflow, or any website other than the Java API, is a no-no.
  - **Using ChatGPT, or any other AI tool, to help you write a program is a no-no.**
  - Using ChatGPT to help you on your writing assignments *is* permitted, but **only in the specific way I will describe in class**. (Spoiler: this will involve a documented draft-and-revision cycle, all parts of which you will turn in.)

All of the above are Honor Code violations.

3. For the *team* activities, you are not only permitted, but mandated, to work with your team openly, frequently, and without reservation. For the *team* activities, it is also permissible to Google for code snippets or design ideas, but **not** to use ChatGPT or any other AI tool.

## A word on procrastination

Procrastination will kill you. Do. Not. Do. It.

There are going to be long stretches this semester (2 weeks or more) during which you are expected to be working towards the next assignment or project milestone. Leaving the whole project until the day or two before the deadline is a recipe for disaster. **You must figure out a way to get yourself (and your teammates)**

**to work regularly and consistently over the time period allotted.** I can't stress this enough. If you suspect you may have trouble with this, come talk to me so I can help you figure out how to set (and meet) intermediate deadlines.

## Guidelines for class participation

I believe that students learn best when they participate wholeheartedly in all aspects of the learning process. Hence while your grade will not be partially determined by any "class participation score" *per se*, it is very much to your advantage, and very much recommended, that you come to lecture every single class period, and participate fully in it.

## How to reach each other

You are responsible for checking both the class website and your UMW email for announcements on a daily basis! To get a hold of me, come to office hours or e-mail me at [stephen@umw.edu](mailto:stephen@umw.edu).

## Title IX Statement

UMW faculty are committed to supporting students and upholding the University's *Policy on Sexual and Gender Based Harassment and Other Forms of Interpersonal Violence*. Under Title IX and this Policy, discrimination based upon sex or gender is prohibited. If you experience an incident of sex or gender based discrimination, we encourage you to report it. **While you may talk to me, understand that as a "Responsible Employee" of the University, I must report to UMW's Title IX Coordinator what you share.** If you wish to speak to someone confidentially, please contact the confidential resources below. They can connect you with support services and help you explore your options. You may also seek assistance from UMW's Title IX Coordinator; their contact information can be found below. Please visit <http://diversity.umw.edu/title-ix/> to view *UMW's Policy on Sexual and Gender Based Harassment and Other Forms of Interpersonal Violence* and to find further information on support and resources.

### Resources

Ruth Davison, Ph.D.  
Title IX Coordinator  
Lee Hall, Room 401  
540-654-5656  
rdavison@umw.edu

### Confidential Resources

#### *On-Campus*

Talley Center for Counseling Services  
Lee Hall 106, 540-654-1053

Student Health Center  
Lee Hall 112, 540-654-1040

#### *Off-Campus*

Empowerhouse (24-hr hotline)  
540-373-9373

RCASA (24-hr hotline)  
540-371-1666

## Recording Policy

Classroom activities in this course may be recorded by students enrolled in the course for the personal, educational use of that student or for all students presently enrolled in the class only, and may not be further copied, distributed, published or otherwise used for any other purpose without the express written consent of the course instructor. All students are advised that classroom activities may be taped by students for this purpose. Distribution or sale of class recordings is prohibited without the written permission of the instructor and other students who are recorded. Distribution without permission is a violation of copyright law. This policy is consistent with UMW's *Policy on Recording Class and Distribution of Course Materials*.

## Accessibility statement

The Office of Disability Resources has been designated by the university as the primary office to guide, counsel, and assist students with disabilities. If you receive services through the Office of Disability Resources and require accommodations for this class, please provide me a copy of your accommodation letter via email or during a meeting. I encourage you to follow-up with me about your accommodations and needs within this class. I will hold any information you share with me in the strictest confidence unless you give me permission to do otherwise.

If you have not made contact with the Office of Disability Resources and have reasonable accommodation needs, their office is located in Seacobeck 005, phone number is (540) 654-1266 and email is [odr@umw.edu](mailto:odr@umw.edu). The office will require appropriate documentation of disability.



## Basic needs security

Learning effectively and engaging wholly in class is dependent upon our basic security and having our fundamental needs met: having a safe place to sleep at night, regular access to nutritious food, and some assurance of safety. If you have difficulty affording groceries or accessing sufficient food to eat every day, or if you lack a safe and stable place to live, please contact Chris Porter, Assistant Dean of Students, at [cjporter@umw.edu](mailto:cjporter@umw.edu). Additionally, the Gwen Hale Resource Center is a free resource on campus, providing food, toiletries and clothing to any member of our community. It is open Monday, Tuesday and Friday from 1pm-6pm, on the 5th floor (floor A for Attic) of Lee Hall, or [resource@umw.edu](mailto:resource@umw.edu). Finally, you are always welcome to talk with me about needs, if you are comfortable doing so. This will enable me to provide any resources I may possess.

## Road map

Week	Topics
1	Motivating OOA&D; CLI development environments
2	OO principles; memory diagrams
3	Encapsulation; UML class diagrams
4	Exception handling; the Singleton pattern
5	UML sequence diagrams
6	Persistence & hydration
7	Polymorphism and inheritance
8	<i>(Spring break)</i>
9	Java odds 'n' ends; Factory pattern
10	Team software development; discovering the design
11	Software requirements: Use Cases
12	Documenting an API
13	Design patterns
14	Introduction to parallelism and concurrency
15	Introduction to parallelism and concurrency