

- [Course Introduction](#)
- [Course Schedule](#)
- [Late Policies](#)
- [Grading Policies](#)
- [Tools for this course](#)
- [Where to go for help? \(Comm. Policies & Methods\)](#)

## About

---

In this class you will have the opportunity to learn the basic skills needed to

1. design and implement a simple to moderately complex database,
2. make good decisions regarding database design, and
3. document your design and queries in standardized notation.

Additionally, you will have the opportunity to practice these skills by building a simple website driven by a database backend.

## Measurable Student Learning Outcomes:

---

At the completion of the course, students will be able to...

1. Describe the difference between a relational database and a flat file (Level 1; ABET Outcomes: A, j)
2. Model a moderately complex data set by using an ER or UML diagram, and derive a relational schema from that diagram (Level 3; ABET Outcomes: A, b, C)
3. Create a relational database from a relational schema (Level 4; ABET Outcomes: A, K, c)
4. Create multiple indices in a relational database, and explain when and why such indices are appropriate (Level 5; ABET Outcomes: A, b, C)
5. Formulate SQL statements for data manipulation (Level 4; ABET Outcomes: cA, c)
6. Formulate simple queries in relational algebra by using projection, selection, product, and join operations (Level 3; ABET Outcomes: A, I)
7. Describe the components and interfaces of a Web-based database system (Level 1; ABET Outcomes: A, B, I)
8. Design and implement a Web-based relational database system, using one or more scripting languages (e.g., PHP) and an open-source database development system (e.g., MySQL) (Level 4; ABET Outcomes: a, B, C, I, K)

## Recommended books

---

*Relational Database Design and Implementation* by Jan L Harrington, 2016 Edition is the recommended textbook for CS340-400.

A rough mapping of the current weeks to the chapters of the book is given below

- Chapters 1,3 -- Week 1

- Chapters 4-5 -- Weeks 2-3
- Chapter 10,16-17, 19 -- Week 4
- Chapter 11 -- Week 5
- Chapters 7, 13-15 -- Week 6
- Chapter 6 -- Week 7

This book is available for free *online* for all the OSU students at the [OSU Library \(Links to an external site.\)](#)

If you decide to use Python for web development in this course, the [Flask user guide \(Links to an external site.\)](#) should be good enough.

If you decide to use node.js for web development in this course, recommended books from CS 290 should suffice.

## Course Content

This course is dedicated to learning the basics of database design and use.

To accomplish this, along with 4 assignments you will work in groups on a term Project implementing the concepts that you learn each week.

There will also be small quizzes which will help you assess your own understanding of the material. With few exceptions, if you ever get a wrong answer on a quiz that means you are not understanding critical information and it is your responsibility to get clarification. I hope you will not hesitate to ask a question on Piazza if this happens!

All the content, assignments, quizzes and project steps will be available in weekly Modules.

## Expectations from Student

### Prior Knowledge

---

Students are expected to know the following:

You should be familiar with good coding practices. Good coding style is required and not taught in this class. You should understand basic control structures. If you are unable to code a simple sorting algorithm, you will have trouble in this class as this level of coding experience is expected.

You should also have completed 290 or be a strong student currently taking the course. It is possible to take these two courses at the same time, but if you run into difficulty in 290 you can have some real trouble late in this course, so do that with some caution.

### Code quality

---

Code must be clear and you must understand what it is doing. Having well-documented code is going to be extremely important. I or the TA may not know the platform you are using, so it is

your responsibility to make sure that your work is clear enough so that we can follow what is happening.

You should also have no major errors in your program. If we can manage to get the program to throw some default error message that is usually a major issue. Errors which you handle via a clear message to the user (e.g. 'Please enter only numbers in the age field') are usually fine. On the other hand, error messages like 'Error 0x00001: Null pointer to Null found, expected pointer to Int Factory Factory' are not.

When possible, you should find a style guide and conform to it.

## **Expectations for Student Conduct**

---

Student conduct is governed by the university's policies, as explained in the [Student Conduct Code](#). Students are expected to conduct themselves in the course (e.g., on discussion boards, email postings) in compliance with the university's regulations regarding civility.

## **About the Instructor Danielle Safonte**

---

Hi, I am Danielle Safonte and will be your instructor this term. I am teaching this course with fellow instructor and Dr. Michael Curry. Catch us on Teams and Piazza!

I live on the east coast in New Jersey, recently moved from Brooklyn, NY. I have been a professor both for on campus courses and online courses at NYC College of Technology and Brooklyn College for many years now. I am excited to join the Oregon State team with you here online.

My undergraduate education was based in Business Information Technology Systems and Software Engineering obtained at the City College of NY. My Masters education was focused in Information Systems and Databases, obtained online in a program similar to this. As both a professor and a student in online education, I hope to bring an enjoyable, informative distance learning experience.

I started in the technology field working for the City University of NY as a lab technician and worked my way up while continuing my education to the Director of Technology and Database Administrator for some select departments in the college. I have created databases from scratch as well as worked with boxed software for customized database and analytic tools.

Through my almost 10 years of creating user manuals and holding training courses, I decided to take on a couple classes to teach. Little did I know a couple short years later, teaching became my main focus. I enjoy bringing real world knowledge into my courses to prepare you for a successful career ahead. I also freelance with a publisher to creating course content for online education. We have a variety of clients including Universities to large corporations, who look for training materials and courses just like this one.

I want you all to take away the most you can from this course. The only silly questions are the ones you walk away with never inquiring for an answer. If you are having a hard time or are unsure of something, please JUST ASK! I am available through Teams (@safonted), Piazza, Zoom or by email. ([danielle.safonte@oregonstate.edu](mailto:danielle.safonte@oregonstate.edu))

In my spare time, I enjoy traveling and spending time with my family. I look forward to connecting with you all and starting this learning journey with you.

## About the Instructor Michael Curry

---

Hi, I am Dr. Michael Curry and will be your instructor this term. I am teaching this course with fellow instructor and Database expert Danielle Safonte. Catch her on Microsoft Teams and Piazza!

I live in Portland Oregon area and have been a professor on campus at Oregon State and now online.

I didn't plan to become a computer scientist, but maybe like you I had a bit of a knack for it and kind of enjoyed doing things that required writing little programs. After college I was a telecommunications officer in the US Army and really didn't write code, but later I went to graduate school and got a job as a software engineer. I initially wrote software for embedded systems but what I didn't like about that was how hard it was to update an embedded system.

So instead I started writing software for the web and I have always liked the interactive nature of the web and especially how easily the code can be updated on the server. I enjoyed it so much I started my own company in 2003 and over the next 12 years worked with over 100 different businesses which was really rewarding to help my clients by writing web software, managing their data and solving problems.

I have been working with databases and writing web code for over 15 years, but importantly for you I have also been fortunate enough to teach this material to students such as yourselves for over 10 years too. And I really enjoy helping others learn. In fact my motto is "when you learn I win". So if you're stuck and having difficulty then I am losing! So don't hesitate to reach out to me in Teams, Piazza or by email.

Outside of work, I am an avid rock climber and love being outdoors. I look forward to connecting with you and helping you learn this term.

## Student Evaluation of Courses

---

The online Student Evaluation of Teaching system opens to students during the week before finals and closes the Monday following the end of finals. Students receive notification, instructions and the link through their ONID. They may also log into the system via Online Services. Course evaluation results are extremely important and used to help improve courses and the online learning experience for future students. Responses are anonymous (unless a student chooses to "sign" their comments, agreeing to relinquish anonymity) and unavailable to instructors until after grades have been posted. The results of scaled questions and signed comments go to both the instructor and their unit head/supervisor. Anonymous (unsigned) comments go to the instructor only.

See the table below for more details on assignments and due dates.

You can also download the syllabus as a pdf.

You can take the [Syllabus Quiz](#) once you are registered for the course.

Week	Topic & Content	Due Mon	Due Thurs
1	Intro/Tools  Relational DB & Design	Syllabus Quiz  Assignment 0  Step 0	Task 0  Quiz 1
2	Models, Diagrams, Schemas	Project Step 1 Final (Proposal and Outline)	Quiz 2
3	SQL (DML)	Project Step 2 Draft (ERD Schema)	Project Step 2 Review  Quiz 3
4	SQL, Advanced DML, DDL	Project Step 2 Final (ERD Schema)	Advanced SQL Assignment  Quiz 4
5	ER to DDL  Relational Algebra	Project Step 3 Draft (HTML Interface)	Project Step 3 Review  Quiz 5
6	Project, ORMs	Project Step 4 Draft (DML + DDL)	Project Step 4 Review
7	Non-relational Databases, Stored Procedures, and Triggers	Project Step 5 & 6 (CREATE + READ + UPDATE + DELETE) Draft	Project Step 5 & 6 Review
8	Final Week	Reflection EC	Project Step 7 FINAL PROJECT DUE

Grade letter	Percentage floor
A	92
A-	90
B+	87
B	82
B-	80
C+	78
C	72
C-	70
D+	68
D	62
D-	60
F	0

## **Grading Scale**

---

The final grade will be computed using a weighted average according to the table above and the weighting below.

### *Note about Partial Credit*

In general assignments are not graded such that each piece is worth a fixed portion of the total credit for the assignment. For example, if 40% of the program is non-functional that generally indicates that there is some important concept that has been missed. If that is the case, then the grade will be a 0. You can then resubmit it to earn up to 70% of the assignment credit, but the revised submission must be entirely functional.

## **Maximizing Credit**

---

You should treat me as a client. I will give you a set of requirements. Requirements can be interpreted differently; this is unavoidable. You have to meet my interpretation of the requirements to get an A. If you have any doubt about what a requirement means, you should ask me to clarify.

Even if you do not plan to do work on the assignment till near the deadline, you should read the requirements as soon as you can to see if anything is ambiguous so you can get clarification while there is still time to act on it.

## Working In Groups

---

Working effectively as part of a team is a requirement for this course. For any group assignment, if you do not submit as a group then you will not receive credit.

Please note that in a group project all group owners individually own *all* rights, with the caveat that they may owe royalties to others if they individually make money on the code later. See this page for more information on ownership or group projects:

[https://fairuse.stanford.edu/overview/faqs/copyright-ownership/#who\\_owns\\_the\\_copyright\\_in\\_a\\_joint\\_work](https://fairuse.stanford.edu/overview/faqs/copyright-ownership/#who_owns_the_copyright_in_a_joint_work)

## Grade Weighting

---

- Quizzes - 15%
- Assignments - 25%
- Participation - 20%
- Project - 40%

## Academic Integrity

---

The Code of Student Conduct prohibits Academic Misconduct and defines it as:

Any action that misrepresents a student or group's work, knowledge, or achievement, provides a potential or actual inequitable advantage, or compromises the integrity of the educational process.

To support understanding of what can be included in this definition, the Code further classifies and describes examples of Academic Misconduct, as follows.

Prohibited behaviors include, but are not limited to doing or attempting the following actions:

- Cheating. Unauthorized assistance, or access to or use of unauthorized materials, information, tools, or study aids. Examples include, but are not limited to, unauthorized collaboration or copying on a test or assignment, using prohibited materials and texts, unapproved use of cell phones, internet, or other electronic devices, etc.
- Plagiarism. Representing the words or ideas of another person or presenting someone else's words, data, expressed ideas, or artistry as one's own. Examples include, but are not limited to, presenting someone else's opinions and theories as one's own, using another person's work or words (including unpublished material) without appropriate source documentation or citation, working jointly on a project and then submitting it as one's own, etc.
- Falsification. Fabrication or invention of any information. Examples include, but are not limited to, falsifying research, inventing or falsely altering data, citing fictitious references, falsely recording or reporting attendance, hours, or engagement in activities such as internships, externships, field experiences, clinical activities, etc.

- **Assisting.** Any action that helps another engage in academic misconduct. Examples include, but are not limited to, providing materials or assistance without approval, altering someone's work, grades or academic records, taking a test/doing an assignment for someone else, compelling acquisition, selling, bribing, paying or accepting payment for academic work or assistance that contributes to academic misconduct, etc.
- **Tampering.** Interfering with an instructor's evaluation of work by altering materials or documents, tampering with evaluation tools, or other means of interfering.
- **Multiple submissions of work.** Using or submitting work completed for another or previous class or requirement, without appropriate disclosure, citation, and instructor approval.
- **Unauthorized recording and use.** Recording and/or dissemination of instructional content without the express permission of the instructor(s), or an approved accommodation coordinated via Disability Access Services.

To support understanding of what can be included in this definition, the Code further classifies and describes examples of Academic Misconduct, including cheating, plagiarism, assisting and others. See the [Code of Student Conduct](#) for details.

You are expected to do your own work and demonstrate academic integrity in every aspect of this course. Familiarize yourself with the standards set forth in the OSU Code of Student Conduct Section 4.2. You must only access sources and resources authorized by the instructor. You may not show your work to any other current or future students without the instructor's authorization. Violations of these expectations or the Code of Student Conduct will be reported to the Office of Student Conduct and Community Standards. If there is any question about whether an act constitutes academic misconduct, it is your responsibility to seek clarification and approval from the instructor prior to acting.

### **Questions about grades**

---

The TAs will grade all your submissions, unless otherwise noted. If you have any concerns, contact **the grader for that specific assignment via email** as noted in the Where to go for help?. Any concerns about grades should be communicated within 7 days of receiving the grade.

### **Statement Regarding Students with Disabilities**

---

Accommodations for students with disabilities are determined and approved by Disability Access Services (DAS). If you, as a student, believe you are eligible for accommodations but have not obtained approval, please contact DAS immediately at 541-737-4098 or at <http://ds.oregonstate.edu>. DAS notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations. While not required, students and faculty members are encouraged to discuss details of the implementation of individual accommodations.

## Accessibility of Course Materials

---

All materials used in this course are accessible. If you require accommodations please contact [Disability Access Services \(DAS\)](#).

Additionally, Canvas, the learning management system through which this course is offered, provides a [detailed page](#) about how the platform is accessible to students with disabilities.

## Late Policy

---

**You must submit all assignments before the due date, even if it is incomplete!** However, if your submission is incomplete your grade maybe a Zero, unless you follow these instructions.

1. Notify the TA that you plan to resubmit it within 3 **days** (you can notify them by email or in the Canvas comment with your submission).
2. If you submit a partial fulfillment of the requirements before the due date and then resubmit a revision within 3 days, your grade will reflect your final submission. If you do not resubmit within 3 days, your grade will reflect your original (potentially incomplete) submission.
3. **The exceptions to this** are project step **drafts**, draft **reviews**, extra credit assignments and the **final project step submission(the one that is due in the last week of the term!)** ; these must be submitted on time and cannot be revised once the due date has passed.
4. **If you do not submit your first version before the due date, you will receive 0 credit.** Exceptions may be made for documented emergencies e.g. hospitalization.

## Communication Methods

---

### Piazza

Piazza should be used for *all* questions seeking technical help, conceptual help or assignment clarification. Essentially if it is any sort of information that may benefit other students, it should be posted to Piazza. If there is a question that gets ignored for longer than 48 weekday hours you can email the instructor and TA and ask them to follow up with it. It may have just gotten missed. Look below for the Code Sharing policy.

### Teams

The Teams channel is the primary mode of conducting office hours by the TAs unless specified. At the beginning and end of each office hour session (whether or not it is held using Teams), the TA would put up a message on the Teams channel informing so. If you are sharing code on Teams, look below at the Code Sharing policy.

### Email

Email should be used for anything that contains sensitive information. So if you have a question about a grade or want to request an extension, do so via email. All emails should have the exact characters (yes, include the parentheses) [CS340] at the start of the subject so that they get priority in my inbox. In addition, every time you reply to an email thread it knocks it to the back of the queue because email gets processed in order

of the most recent emails last. So be careful sending multiple replies as it might bump you back in the queue.

#### Canvas Comments

Canvas comments should *only* be used by students to add commentary *prior* to grading and by myself and the TAs to give feedback. If you need to communicate something to myself or the TAs after your assignment has been graded do so via Email. We *will not* see comments posted to your assignment submission after it has been graded.

#### Canvas Mail

The TAs and myself try the best we can to monitor Canvas email. But there are less options to sort and filter mail via that system so there is a much higher likely-hood that we will miss things that get sent via Canvas mail. So I suggest you do *not* use it.

### **TA Office Hours**

---

The primary mode of office hours by the TAs, unless otherwise indicated, will be the Teams channel. At the beginning and end of each office hour, the TAs will inform on the above Teams channel.

Times are in PDT.

**Monday**                      TBD

---

**Tuesday**

---

**Wednesday**

---

**Thursday**

---

**Friday**

#### **Contacting the TAs**

- Bhavya Parikh [parikhb@oregonstate.edu](mailto:parikhb@oregonstate.edu)
- Rashmi Suresh Jadhav [jadhavr@oregonstate.edu](mailto:jadhavr@oregonstate.edu)
- Yichaun Yin [yinyic@oregonstate.edu](mailto:yinyic@oregonstate.edu)
- George Kochera [kocherag@oregonstate.edu](mailto:kocherag@oregonstate.edu)
- Gerrit Van Ruiswyk [vanruisg@oregonstate.edu](mailto:vanruisg@oregonstate.edu)
- Gabrielle Pang [pangg@oregonstate.edu](mailto:pangg@oregonstate.edu)
- Paul Newling [newlingp@oregonstate.edu](mailto:newlingp@oregonstate.edu)
- Rebekah Koon [koonr@oregonstate.edu](mailto:koonr@oregonstate.edu)
- Andrew Jung [jungan@oregonstate.edu](mailto:jungan@oregonstate.edu)

### **Contacting the Instructor**

---

Office hours with the Instructor are available by appointment.

You can email Danielle Safonte at [danielle.safonte@oregonstate.edu](mailto:danielle.safonte@oregonstate.edu)

Or Michael Curry at [michale.curry@oregonstate.edu](mailto:michale.curry@oregonstate.edu)