San José State University
College of Science/Department of Computer Science
CS 152, Programming Paradigms, Section 05, Fall 2020

Course and Contact Information

Instructor(s):	Michael McThrow
Office Location:	Online via Zoom
Email:	michael.mcthrow@sjsu.edu
Office Hours:	Wednesday 6:15pm-7:15pm PDT/PST (online via Zoom) Also available via appointment.
Class Days/Time:	Monday/Wednesday 7:30pm-8:45pm PDT/PST
Classroom:	https://sjsu.instructure.com/courses/1377555
Prerequisites:	CS 151 or CMPE 135 (with a grade of "C-" or better) **This prerequisite will be strictly enforced.** Course is restricted to Computer Science, Applied and Computational Math, or Software Engineering majors except with instructor's consent.

Course Description

(Copied from http://info.sjsu.edu/web-dbgen/catalog/courses/CS152.html) Programming language syntax and semantics. Data types and type checking. Scope, bindings, and environments. Functional and logic programming paradigms, and comparison to other paradigms. Extensive coverage of a functional language.

Course Format

CS 152 is an online course this semester. All instruction will be delivered online, and all homework assignments and exams will be turned in electronically. I will be delivering online lectures via Zoom, and these lectures will be recorded for later review and for the convenience of students who cannot attend class at the scheduled time. Since these are live lectures, I will be able to respond to students' questions in real time.

All students must have access to a personal computer and have reliable Internet access in order to participate in this course.

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on the Canvas Learning Management System course login website at http://sjsu.instructure.com. You are responsible for regularly checking with the messaging system through MySJSU on Spartan App Portal to learn of any updates.

Course Goals

- To study programming language design, with an emphasis on the functional paradigm.

Course Learning Outcomes (CLO)

- To ensure that students gain an understanding of programming language design and language translation.
- To achieve competence in a functional programming language.

Upon successful completion of this course, students will be able to:

1. Have a basic knowledge of the history of programming languages.
2. Have a basic knowledge of the procedural, object-oriented, functional, and logic programming paradigms.
3. Understand the roles of interpreters, compilers, and virtual machines.
4. Critique the design of a programming language.
5. Read and produce context-free grammars.
6. Write recursive-descent parsers for simple languages, by hand or with a parser generator.
7. Understand variable scoping and lifetimes.
8. Write interpreters for simple languages that involve arithmetic expressions, bindings of values to names, and function calls.
9. Understand type systems.
10. Understand the implementation of procedure calls and stack frames.
11. Produce programs in a functional programming language in excess of 200 LOC.

Required Texts/Readings

Textbooks

- (SICP) Ableson, Sussman, and Sussman. *Structure and Interpretation of Computer Programs, 2nd Edition*, 1996. Available online for free at https://mitpress.mit.edu/sites/default/files/sicp/index.html.
- (TYSiFD) Dorai Sitaram. *Teach Yourself Scheme in Fixnum Days.* 2015. Available online for free at https://ds26gte.github.io/tyscheme/.
- (TAoP) Sterling and Shapiro. *The Art of Prolog, Second Edition: Advanced Programming Techniques*, 1993. Available online at https://mitpress.mit.edu/books/art-prolog-second-edition; click on "Open Access" and then click "Download PDF" in order to download a free copy of the book.
- (S80) Goldberg and Robson. *Smalltalk-80: The Language and Its Implementation*. 1983. Available online for free at http://stephane.ducasse.free.fr/FreeBooks/BlueBook/Bluebook.pdf.

Other Readings

See the "Readings" column in the Course Schedule below. Periodically I will be assigning academic papers, web pages, and handouts for additional required reading material.

Note that the readings may be subject to change; I will notify my students of any changes in the required readings.

Other technology requirements / equipment / material

- A personal computer with reliable Internet access.
- Access to a terminal connected to a system running a Unix-like operating system (Windows Subsystem for Linux on Windows 10 meets this requirement).

Course Requirements and Assignments

"Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction,

preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus."

Final Examination or Evaluation

There will be a take-home final exam assigned during finals week. The final exam will be cumulative.

Grading Information

Each student's grade is computed based on the following composition:
Labs: 20%
Projects: 40%
Exams (one midterm, one final): 40% (midterm worth 20%, final worth 20%)

Class attendance is not mandatory. However, any topic taught during the lectures may show up on the exams even if it was not covered in the required readings. The reverse is also true; any material covered in the assigned readings or in the labs and projects may show up on the exams even if it wasn't mentioned during the lectures. Additionally, the live classes will provide opportunities for me to answer any questions you have about the course. Nevertheless, if you cannot attend class during the scheduled time, then please watch the recorded lectures.

Below is the late work policy for this course:

- Deadlines for labs, projects, and take-home exams will be at 11:59:59pm Pacific Daylight Time (or Pacific Standard Time starting November 1) on the due date.
- For labs and projects (not exams), if a student turns in an assignment after the deadline, the student has three additional days to turn in the assignment with a 10% late penalty. I will not be accepting any assignments after the additional three-day period except under extenuating circumstances.
- All take-home exams must be turned in no later than the due date unless there are extenuating circumstances.
- Students with extenuating circumstances, such as a medical emergency, should contact me as soon as possible.

Unless noted otherwise, **all labs, projects, and exams must be individual work**. Students may not collaborate on these tasks, though they may study in groups and they may share graded work with each other. **Sharing solutions to labs and projects or posting them online before they have been graded is prohibited. Sharing exam questions or posting them online before exams have been graded is prohibited.** After grades have been issued for any lab, project, or exam, students may share the questions and answers. Any student violating these rules will be subject to discipline as described in University Policy F15-7, which can be found at https://www.sjsu.edu/senate/docs/F15-7.pdf.

Below is the grading scale for this course:

Grade	Minimum Threshold
A plus	97%
A	93%
A minus	90%
B plus	87%
B	83%
B minus	80%
C plus	77%
C	73%

Grade	Minimum Threshold
C minus	70%
D plus	67%
D	63%
D minus	60%
F	0%

Classroom Protocol

- During live Zoom lectures, please keep your microphones muted during instruction, except when I am taking questions.
- Please be respectful in all communication in this course. This includes the lectures and all online communication.
- Please keep up with all communication in this course, whether it is through email or through Canvas.

University Policies

Per University Policy S16-9, relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo), which is hosted by the Office of Undergraduate Education.

CS 152 / Programming Paradigms, Fall 2020, Course Schedule

Please note that I reserve the right to change the course schedule. I will communicate all changes to the course schedule via Canvas and email.

Course Schedule

Here is a rough overview of the topics covered:

- Overview of Course (Week 1)
- Procedural and Structured Programming (Week 2)
- Syntax and Semantics (Weeks 2 and 3)
- Functional Programming, Scheme, Interpretation, and Compilation (Weeks 3-7)
- Type Systems (Week 9)
- Logic Programming and Prolog (Weeks 10-13)
- Object-Oriented Programming Revisited (Weeks 14 and 15)
- Miscellany (Week 16)

Week	Date	Topics	Readings (before class)	Assignments with Deadlines
1	8/19	Cancelled Due to Fires and Smoke		
2	8/24	Introduction to CS 152, Overview of Course, Procedural Programming and Structured Programming	"Go To Statement Considered Harmful" by Edsger W. Dijkstra (1968)	Proof of Prerequisite assigned
2	8/26	Parsing, Context-Free Grammars, and Abstract Syntax Trees	Handout (will provide on 8/24)	Lab #1 (Infix and Postfix Calculators) assigned **Proof of Prerequisite Due**
3	8/31	Operational Semantics and the Lambda Calculus (**Note:** Last day to drop course without a W grade.)	Handout (will provide on 8/26)	
3	9/2	Lab #1 Review		
4	9/7	**HOLIDAY – LABOR DAY** (**Note:** September 8 is the last day to add courses and to register late)		**Lab #1 Due**
4	9/9	Functional Programming and An Introduction to the Scheme Programming Language	TYSiFD Chapters 1-5 Optional: SICP Chapter 1	Lab #2 (Scheme) Assigned
5	9/14	An Introduction to the Scheme Programming Language	TYSiFD Chapters 6-7 Optional: SICP Chapter 2.	
5	9/16	State, Mutation, and Environments	SICP Chapter 3 (from 3 to 3.3.5)	Project #1 Assigned
6	9/21	Building an Evaluator	SICP Chapter 4 (from 4 to 4.1.7)	**Lab #2 Due**

Week	Date	Topics	Readings (before class)	Assignments with Deadlines
6	9/23	Building an Evaluator		
7	9/28	Virtual Machines and Compilation	SICP Chapter 5-5.5	
7	9/30	More Advanced Scheme Features: Continuations and Macros	TYSiFD Chapters 8 and 13	**Project #1 Due on 10/2**
8	10/5	Review for Midterm Exam		
8	10/7	**MIDTERM EXAM**		
9	10/12	Introduction to Type Systems	Handout (will provide on 10/7)	Project #2 Assigned Lab #4 (Type Systems) Assigned
9	10/14	Introduction to Type Systems		
10	10/19	Introduction to Logic Programming	TAoP Introduction, Chapters 1 and 2	**Lab #4 Due** Lab #5 (Logic Programming) Assigned
10	10/21	Introduction to Logic Programming	TAoP Chapter 3	
11	10/26	Resolution and Unification in Logic Programming	TAoP Chapters 4 and 5	**Lab #5 Due**
11	10/28	Introduction to Prolog	TAoP Chapters 6-8	
12	11/2	Introduction to Prolog	TAoP Chapters 9 and 10	Lab #6 (Prolog) Assigned
12	11/4	Cuts and Negation in Prolog	TAoP Chapter 11	
13	11/9	Applications of Prolog: A Game and a Compiler	TAoP Chapters 21 and 24	
13	11/11	**HOLIDAY – VETERANS DAY**		
14	11/16	Introduction to Smalltalk and Message-Passing Object-Oriented Programming	S80 Chapters 1-4	**Project #2 Due** **Lab #6 Due** Lab #7 (Smalltalk) Assigned Project #3 Assigned
14	11/18	Metaprogramming in Smalltalk	S80 Chapters 5-6	
15	11/23	The Self and JavaScript Programming Languages	"SELF: The Power of Simplicity" by David Ungar and Randall B. Smith (1987)	**Lab #7 Due**
15	11/25	**HOLIDAY – THANKSGIVING BREAK**		
16	11/30	Miscellaneous Topics in Programming Languages	To be determined (will provide 11/23)	
16	12/2	Miscellaneous Topics in Programming Languages	To be determined (will provide 11/30)	**Project #3 Due on 12/4**

Week	Date	Topics	Readings (before class)	Assignments with Deadlines
17	12/7	Review for Final Exam		
17	12/9	**FINAL EXAM**		