San José State University
Department of Computer Science
CS 152, Programming Language Paradigms, Section 4, Fall 2021

Instructor(s):	Tazmina Sharmin
Email:	tazmina.sharmin@sjsu.edu
Office Hours:	Tuesday 11 - 12 pm
Class Days/Time:	Tuesday 9 - 10:15 am | Thursday 9 - 10: 15 am
Classroom:	Online via Zoom
Prerequisites:	CS 151 or CMPE 135 (with a grade of "C-" or better)

Course Description

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

The course will be conducted online over Zoom. All instruction will be delivered online, and all homework assignments and exams will be turned in electronically. All students must have access to a personal computer and have reliable Internet access in order to participate in this course.

Canvas Course Site

Course materials such as syllabus, lecture notes, assignments and exams can be found on the Canvas Leaning Management System course website at http://sjsu.instructure.com/. You are responsible for regularly checking with Canvas to learn of any updates.

Program Information (Delete if not applicable)

Insert any program or department information here as appropriate.

Course Goals

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

1. Understand programming language design.

2. Achieve competence in a functional programming language.

Course Learning Outcomes (CLO)

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.

Textbook

1. (TYSiFD) Dorai Sitaram. Teach Yourself Scheme in Fixnum Days. 2015. Available online for free at https://ds26gte.github.io/tyscheme/.
2. • (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.

   Additional Readings TBD.

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."

Homework assignments will be posted and submitted on Canvas. For full credit, they must be submitted by the posted due date and time. All assignments will be individual work. You may not share or copy code or answers from fellow students or from the web. If someone else copies your work, with or without your permission, you will be held responsible.

There will be one team project. You may choose to work by yourself or with a partner. For a team project, the work must be done by both team members and both team members will receive the same grade.

There is a final and a midterm. The final exam is worth 20% of the total grade for the class.

Grading Information

Homework: 30%
Project: 30%
Midterm: 20%
Final: 20%

Late Work

Late Work Late assignments will be accepted with a 1-point penalty for each day late. Late days include weekend days. For example, an assignment due on Monday by 5 PM will incur a penalty of 1 point if submitted at 8AM on Tuesday. No submissions will be accepted more than 3 days late.

Grade Scale

The letter grade will be determined based on the following scale:
A+ = 98% - 100%
A = 92% - 97%
A- = 90% - 91%
B+ = 87% - 89%
B = 83% - 86%
B- = 80% - 82%
C+ = 77% - 79%
C = 73% - 76%
C- = 70% - 72%
D = 60% - 69%
F = below 59

Classroom Protocol

- During live Zoom lectures, please keep your microphones muted during instruction, except when I am taking questions or while I am hosting a discussion.
- 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.
- Limit your distractions/avoid multitasking. You can make it easier to focus on the meeting by turning off notifications, closing or minimizing running apps, and putting your smartphone away (unless you are using it to access Zoom).

  Attendance is recommended, but it is not mandatory, except for exam dates.

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/ . Make sure to review these policies and resources.

Important dates:

Tuesday, August 31: Last day to drop courses without a "W" grade

Tuesday, September 08: Last day to add classes, register late, and submit Credit/No Credit(CR/NC) request

CS 152 Programming Paradigms, Fall 2021, 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.

Week	Date	Topics	Homework
1	8/19/2021	Course introduction	
2	8/24/2021	Procedural Programming and Structured Programming	
2	8/26/2021	Parsing, Context-Free Grammars, and Abstract Syntax Trees	HW1 assigned
3	8/31/2021	Parsing with ANTLR	
3	9/2/2021	Operational semantics and Lambda Calculus	HW1 due, HW2 assigned
4	9/7/2021	Functional Programming and An Introduction to the Scheme Programming Language	
4	9/9/2021	Scheme: higher order functions, recursion and efficiency	HW2 due, HW3 assigned
5	9/14/2021	Closures and Scoping, Macros	
5	9/16/2021	Building an Interpreter	HW3 due, HW4 assigned
6	9/21/2021	Functional Data Abstraction	
6	9/23/2021	Memory Management	HW4 due, HW5 assigned
7	9/28/2021	Object-Oriented Programming	
7	9/30/2021	Inheritance and Polymorphism	HW5 due
8	10/5/2021	Type Systems, Static and Dynamic Typing	
8	10/7/2021	Midterm Review	
9	10/12/2021	Midterm	

9	10/14/2021	Introduction to Logic Programming	HW6 assigned
10	10/19/2021	Prolog	
10	10/21/2021	Resolution and Unification in Prolog	HW6 due, HW7 assigned
11	10/26/2021	Lists in Prolog and debugging Prolog	
11	10/28/2021	Template Metaprogramming	HW7 due, HW8 assigned
12	11/2/2021	Introduction to Smalltalk	
12	11/4/2021	Metaprogramming in Smalltalk	HW8 due, HW9 assigned
13	11/9/2021	Ruby	
13	11/11/2021	Ruby Blocks	HW9 due, HW10 assigned
14	11/16/2021	Virtual Machines	
14	11/18/2021	Dynamic code evaluation, taint analysis, and information flow analysis	HW11 due, HW12 assigned
15	11/23/2021	Python and IDE plugins	
15	11/25/2021	Thanksgiving	
16	11/30/2021	Advanced Java Features	HW12 due
16	12/2/2021	Final Review	
17	12/9/2021	Final	