San José State University

College of Science/Department of Computer Science CS152,
Programming Language Paradigms, Section 5, Fall, 2020

Course and Contact Information

Instructor:

Linsey Pang (Xiaolin Pang)
Office Location:

Telephone:

Email: xiaolin.pang@sjsu.edu
Office Hours: by appointment

Class Days/Time: Tuesday 7:30-8:45pm | Thursday 7:30-8:45pm

Classroom: online

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. Prerequisite: CS 151 or CMPE 135 (with a grade of
"C-" or better); Computer Science, Applied and Computational Math or Software Engineering
Majors only; or instructor consent.

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. Understand 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.

Required Texts/Readings Textbook

NA

Course Requirements and Assignments

- University Policy S16-9: "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."

- Team project – design and implement an application (topics will be given later in the semester). Apply object-oriented design process (CRC and UML), design patterns, GUI programming and concurrency (detailed requirements will be presented at the lecture). Documentation and collaboration tools required.

Final Examination or Evaluation

Final exam with multiple choice questions and short answer questions and questions that require pseudocode and/or computations.

Grading Information Determination of Grades

- Total points for the course will be weighted by:

  - Exams: 20%
  - Final Exam: 30%
  - Lab/Quiz: 5%
  - Assignments 25%
  - Project 20% (Presentation Required)

- Letter grades will be assigned according to the following policy:

Percentage	Grade
100 -99	A+
93 – 98	A
89 -- 92	A-
87 -- 88	B+
83 -- 86	B

80 -- 82	B-
77 -- 79	C+
73 -- 76	C
70 -- 72	C-
67--69	D+
63 -- 66	D
60 -- 62	D-
59 below	F

- No late work accepted or make-ups.

- Classroom Protocol

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

    - Please bring your laptop if possible

    - Behavior: eating, personal loud discussions, cell phones, laptops are not allowed in the classroom. Skateboards are not allowed inside or outside the classroom.

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/.

• Policy on Academic Integrity
"Your commitment, as a student, to learning is evidenced by your enrollment at San Jose State University. The University Academic Integrity Policy F15-7 requires you to be honest in all your academic course work. Faculty members are required to report all infractions to the office of Student Conduct and Ethical Development. Visit the Student Conduct and Ethical Development website for more information."

Please note that this schedule is subject to change with in-advance notice. Any changes will be announced in class and posted on the Canvas course site.

Tentative Course Schedule

Week	Lecture#	Date	Topics
1	1	08/20/20	Introduction
2	2	08/25/20	Historical overview, computational paradigms
2	3	08/27/20	Language Design Criteria
3	4	09/01/20	Functional Programming
3	5	09/03/20	Functional Programming

4	6	09/08/20	Lists processing
4	7	09/10/20	Higher order functions
5	8	09/15/20	JavaScript
5	9	09/17/20	JavaScript
6	10	09/22/20	Logic Programming
6	11	09/24/20	Review Exam1
7	12	09/29/20	Exam1
7	13	10/01/20	Language translation
8	14	10/06/20	Lexical analysis and parsing
8	15	10/08/20	Lexical analysis and parsing
9	16	10/13/20	Operational semantics
9	17	10/15/20	Compiling
10	18	10/20/20	Compiling
10	19	10/22/20	Data Types
11	20	10/27/20	Data Types
11	21	10/29/20	Review Exam2
12	22	11/03/20	Exam2
12	23	11/05/20	Object-oriented Programming
13	24	11/10/20	Object-oriented Programming
13	25	11/12/20	Python
14	26	11/17/20	Python
14	27	11/19/20	IDE plugins
15	28	11/24/20	Project Presentation
15		11/26/20	Thanksgiving, No class
16	29	12/01/20	Project Presentation
16	30	12/03/20	Review Final -Last Day of Instruction – Last Day of Classes
17	31	12/14/20	Final Exam