# CSc 135 - Computing Theory and Programming Languages

3 Credits
Summer 2023

- *Administrative Information*

  1. **Instructor:** Parham Phoulady
  2. **Department:** Computer Science
  3. **Meeting Time and Location:** Monday/Wednesday, 9:00am - 12:30am – Online
  4. **E-Mail:** `phoulady@csus.edu`

- *Course Description*

  – Introduction to computing theory with examples and applications. Automata and formal languages; regular expressions; deterministic and non-deterministic finite automata; pumping lemma for regular languages; push-down automata and context-free grammars; language recognition; parsing techniques including recursive-descent; Turing machines; computable and non-computable functions. Design and implementation of selected features of programming languages, particularly parameter-passing methods, and approaches for scope and binding. Functional programming paradigm.

- *Prerequisites*

  – CSc 28, CSc 35, and CSc 130 (all with a C- or better).

- *Course Objectives*

  Students completing this course will be able to

  1. Design generators and recognizers for simple regular and context-free languages and give an example of algorithmically converting from one to the other;
  2. Show a context-free grammar ambiguous using parse trees;
  3. Explain the limits of regular and context-free languages and use a pumping lemma to show a language not to be regular and/or context-free;
  4. Discuss parsing and scanning techniques including the recursive-descent or table-driven parsing, derivations, parse trees;
  5. Fundamental concepts of the major programming paradigms and characteristics;
  6. Functional programming and functional languages such as Scheme, LISP, or ML, higher-order functions, lambda functions, and tail-recursion.
  7. Use functional programming languages to solve simple problems; and
  8. Describe the importance of Turing machines, the halting problem, and reductions in the context of the limits of computer algorithms.

- *Textbooks*

  Required: "An Introduction to Formal Languages and Automata", 6th Edition, by Peter Linz, Jones & Bartlett Learning, 2017.

  Other suggested reference(s):

  – "Introduction to the Theory of Computation", 3rd Edition, by Michael Sipser, Cengage Learning, 2013.

- *Grading*
  - *Grading Rubric*
    1. Participation: 5%;
    2. Homework Assignments: 25%;
    3. Midterm: 35%;
    4. Final Exam: 35%;
  - *Grading Scale*
    Your weighted average for the course, using the above rubric, must be greater than or equal to <avg> in order to receive a grade of <grade>.

| <avg> | <grade> |
|---|---|
| 85.0% | A |
| 70.0% | B |
| 55.0% | C |
| 40.0% | D |
| 0% | F |

  - The instructor **reserves the right** to make adjustments to the grading scale based on class averages.
  - The instructor **reserves the right** to make adjustments to students' grades based on their attendance, participation in discussions, or progress.
  - The instructor **reserves the right** to use the +/- grading system.
- *Course Policies & Procedures*
  - All of your submission on Canvas should be sorted in the order of questions and submitted as a singled PDF file. If your solutions are hand-written they must be scanned using a scanner app. Otherwise, they will not be considered.
  - *Exams*
    * Requests for make-up examinations will not be entertained. I will only make exceptions to this policy in case of excused absences. You must provide sufficient documentation to prove that your absence is excused.
    * The date provided for the exam is tentative. Changes to an exam date will be announced in-class at least one week prior to the exam.
    * In case of online submission, late submission will not be accepted.
    * The final exam is cumulative.
  - *Homework Assignments*
    * The penalty for handing in an assignment late is a deduction equal to 25% of the assignment's maximum score per day (not just per weekday when the class is in session; this penalty is also assessed for off-days and holidays). *In Summer classes, this penalty would be 50% per day.*
    * The assignments are due at midnight on the assignment's due date.
    * Homework assignments may be graded by grading a subset of problems. In that case, the grade you get on them will be extrapolated to the whole assignment. For example, let's say a subset of problems are graded that add up to 50 points in total. Then your submission grade will be twice the grade you get on those problems. The main purpose of this is to quicken the grading process.
    * Based on the performance of the whole class, the lowest grade might be dropped.
  - *Regrade Policies*
    * Regrade requests must be submitted, in writing (or by email in case of online classes), to the instructor **within seven calendar days** of either: (1) the date the graded material is returned in class or (2) the date the grades are posted on Canvas, whichever occurs first. *In Summer classes, you should submit your requests within three calendar days.*

    - Regrade requests must be written on a separate sheet of paper and must be attached to your original submission.
    - Regrade requests must specify the question(s) or parts to be regraded. Regrade requests must include a brief description of why the question(s) or parts should be regraded.
    - The instructor reserves the right to regrade the entire submission.
    - Graded material which has been modified in any way since it was returned to the student will not be regraded.
  - *Attendance Policies*
    - Students are expected to **attend all classes**.
    - The participation grades for students are computed based on each student's attendance and the level of participation in discussions in class.
    - You are responsible for all material presented during each lecture. Material presented during the lectures may not be in the textbook or the uploaded notes/slides.
    - Students who anticipate the necessity of being absent from class due to the observation of a major religious observance must provide notice of the date(s) to the instructor, in writing, by the second class meeting.
  - Hand-written work must be legible. If your work is illegible it will not be graded (i.e., you will earn a grade of 0).
  - The instructor reserves the right to interpret the class policies if confusions occur.
  - Avoid excessive collaboration on individual assignments. A healthy amount of discussion between students is a valuable part of learning. But it is not acceptable to collaborate in developing the details of a solution, unless the assignment is a team project.

- *Academic Integrity/Academic Dishonesty*

  Computer Science students are required to adhere to campus and departmental guidelines for academic integrity. Campus guidelines are outlined in the CSUS University Policy Manual on Academic Honesty (`https://www.csus.edu/umanual/student/stu-0100.htm`). The following is additional information specific to Computer Science courses.

  Avoid Cheating!

  - *Avoid excessive collaboration on individual assignments.* A healthy amount of discussion between students is a valuable part of learning. But it is not acceptable to collaborate in developing the details of a solution, unless the assignment is a team project.
  - *Do not give answers to other students.* Giving fellow students answers they were supposed to determine on their own is cheating. Doing so also deprives them of the learning experience of reaching a solution through their own efforts.
  - *Keep all printouts, diskettes, notes, etc. secure*, so that other students will not find them and use your solutions in their work. This also includes shared network drives, printer queues, temporary directories in lab workstations, etc.
  - *Limit discussion to high-level concepts.* Discussing concepts, assignment requirements, syntax errors, coding tricks, or programming environments is generally encouraged. When discussion involves specific code or solutions, it may cross the line into cheating.
  - *Always stop and think before copying or emailing any source code.*
  - *Ask your instructor if you are unsure.* Students who are unsure whether their activities might be considered cheating are encouraged to consult with their instructor. Openness can make a cheating instance considerably less flagrant than if it were covered-up.

  The CSU is committed to providing an electronic environment that is accessible to everyone, including individuals with disabilities. If you encounter any accessibility problems and need assistance, please contact Services to Students with Disabilities: `http://www.csus.edu/sswd/`; `sswd@csus.edu`; (916) 278-6955; (916) 278-7239 (TTY).

- *Tentative Schedule*
  - Lecture 1: Introduction
  - Lecture 2: Languages and Grammars
  - Lecture 3: DFA, Regular Languages and NFA
  - Lecture 4: State Reduction, Regular Expressions and CFL
  - Lecture 5: PDA, BNF and Parse Trees
  - Lecture 6: Functional Programming and Scheme
  - Lecture 7: Chomsky Hierarchy, Pumping Lemma for Regular Languages and CFL
  - Lecture 8: Turing Machine, Turing Thesis and Computability
- *Homework Assignments*
  - Assignment 1: Proofs, Languages and Grammars
  - Assignment 2: Finite Automata and Regular Languages
  - Assignment 3: State Reduction, Regular Expressions and CFL
  - Assignment 4: Programming: Grammars, Parsers and Lexers (with ANTLR 4)
  - Assignment 5: Programming: Functional Programming
  - Assignment 6: Pumping Lemma and Closure Properties
- *Exams*
  - Midterm: Lectures 1–5
  - Final Exam: Cumulative

| Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|
| May 29th | 30st | June 31st **1**<br>Orientation<br>Lecture 1 | 1st<br>**Watch videos of Lecture 1**<br>*Assignment 1 Out* | 2nd<br>Work on test questions on Lecture 1 |
| 5th **2**<br>Lecture 2 | 6th<br>*Assignment 2 Out* | 7th **3**<br>**Watch videos of Lecture 2 and first video of Lecture 3** | 8th<br>Assignment 1 Due | 9th<br>Work on test questions on Lecture 2 |
| 12th **4**<br>Lecture 2<br>Lecture 3 | 13th<br>*Assignment 3 Out* | 14th **5**<br>**Watch videos of Lecture 3 and Lecture 4** | 15th<br>Assignment 2 Due | 16th<br>Work on test questions on Lecture 3 and Lecture 4 |
| 19th **6**<br>Lecture 3<br>Lecture 4 | 20th<br>*Assignment 4 Out*<br>*Assignment 5 Out* | 21st **7**<br>**Watch videos of Lecture 5 and Lecture 6** | 22nd<br>Assignment 3 Due | 23rd<br>Work on test questions on Lecture 5 |
| 26th **8**<br>Lecture 4<br>Lecture 5 | 27th<br>*Assignment 6 Out* | 28h **9**<br>**Watch videos of Lecture 7 and Lecture 8** | 29th<br>Assignment 4 Due<br>Assignment 5 Due | 31st<br>Work on test questions on Lecture 7 and Lecture 8 |
| 3rd **10**<br>Lecture 7<br>Lecture 8 | 4th | 5th **11** | 6th<br>Assignment 6 Due | 7th |

How to study in a typical week during Summer:

- Monday: Attend classes.
- Tuesday: Work on the related lecture's test questions (with no answers).
- Wednesday: Watch the videos and write down your questions.
- Thursday: Work on the related lecture's test questions (with answers).
- Friday: Work on the related assignment.
- Saturday: Work on the related assignment.