CSCI 301 Formal Languages and Functional Programming

Summer 2017

Instructor: Dr. Yudong Liu
Office: CF 483
Email: yudong.liu@wwu.edu
Lectures: MTWR 12-12:50pm, CF 316
Office Hours: MTW 3:30-4:30, and by appointment.
Teaching Assistants
Email:

a. Chace Jones (12-1 :50pm) jonesc48@wwu.edu
b. Nicholas Majeske (4-5:50pm) majeskn@wwu.edu

Lab sessions: Friday 12-1:50pm, 4-5:50pm, CF 416. You may attend the sessions you did not sign up for on a space available basis. Labs start from the first week of class. There are labs the last week of the class.

Course Description: Introduction to discrete structures important to computer science, including sets, trees, functions, and relations. Proof techniques. Introduction to the formal language classes and their machines, including regular languages and finite automata, context free languages and pushdown automata. Turing machines and computability will be introduced. Programming using a functional language is required in the implementation of concepts. Includes lab.

Learning Outcomes: On completion of this course, students will demonstrate:

- Thorough understanding of the mathematical definitions of concepts important to computer science, including sets, tuples, lists, strings, languages, graphs, trees, functions and relations
- The ability to prove basic theorems involving these mathematical concepts
- The ability to employ effectively the functional programming style in a functional programming language
- Solid understanding of fundamental classes of languages, including regular and context free, and their corresponding machines
- Basic understanding of Turing machines and computability
- Basic understanding of important algorithms, including conversion of finite automata to different forms, conversion of grammars to machines
- Basic understanding of LL(k) and LR(K) grammars and the parsing techniques used for those grammars

Goals: This class is an introduction to computer science theory. This is exactly parallel to the distinction between theoretical physics and applied physics. We will use simplified, abstract, ideal mathematical models of computers, so that we can study the theoretical limits of what they can accomplish. We will begin studying the basic mathematics we need, and then study mathematical models of computers of increasing complexity and power. Two classes of computers, finite state automata and push-down automata, also form the basis of powerful programming paradigms useful in a variety of situations. We will also study the functional programming language Scheme. Its simplicity, power, and mathematical elegance will inform our study of computers in the abstract, and also teach us new styles of programing.

Texts:

1. *Book of Proof*. Freely downloadable here.
2. *Introduction to the Theory of Computation*, Freely downloadable here.
3. Course Slides, available on the Canvas site for the course.

Scheme:

- http://www.scheme.com/tspl3/
    - Note: Use the 3rd edition and do not use the 4th edition of the book. Our software is based on R5RS scheme and the 3rd edition covers this. The 4th edition is based on R6RS scheme. The *help desk* within Racket leads to documentation on the implementation.
- Software: ***DrRacket***, available here http://racket-lang.org/

Assessment

1. 6 lab programming assignments (30%).
2. 4 homework assignments (32%).
3. In-class exercises for participation (6%)
4. 6 weekly quizzes (12%)
5. Final Exam (20%)

- The final exam will be on 7/27 during regular lecture hours.
- Each assignment is individual work. You are required to submit your original work.

Grading Policy

Percentage of Points	Grade
90-100	A
80-89	B
70-79	C
60-69	D
< 60	F

The use of "+" and "-" is at my discretion.

Stipulation. There will be graded work due in the last week of the quarter.

Attendance. Attendance is not required but strongly recommended. Studies show that regular attendance is highly correlated with performance.

You are responsible for everything that transpires during class. Make sure to get notes from someone other than me when you are unable to attend.

I will periodically take attendance in class through in-class exercises. Perfect attendance is worth 6 points.

If you have an emergency for one of the exam days, notify me as soon as possible. I will handle each case individually.

Grade Disputes: If you have any questions about a grade you received on homework or an exam, return the homework or exam to me with written questions within 48 hours of the time the graded work is returned to you.

Homework Late Submission Policy: I understand that there might be many reasons for you to turn in late work. However, late work causes enormous management difficulties. My policy therefore be this: A late work could be accepted up to a maximum of 3 days past the deadline but has to take penalty of 10% per day deduction from your score of that assignment. One hour late counts one day late.

Medical Excuse Policy "Students are instructed to contact their professor or teaching assistant in the event they need to miss class due to an illness, injury or an emergency." "In certain circumstances where the illness or injury is prolonged (an absence of more than five days) and requires medical attention or hospitalization we will work with students in providing appropriate documentation."

Cheating. Please review the Computer Science Department's policy on cheating, as the penalties are severe. The department has a strict policy concerning plagiarism. Passing someone else's work off as your own or sharing your code with others results in an F for the course. You are allowed to discuss assignments with other students, but you must not share code. To avoid plagiarism do not show your code to other students and do not read their code. If you copy code from a book, include a comment that gives the appropriate reference.

This policy does not imply that you cannot discuss problems with other students and work things out collaboratively. It implies only that your code and your other written work must not be a direct copy.

Tentative Weekly Schedule (subject to change)

WEEK	DATE	TOPICS	Text book	Assignment
1	6/20 – 6/22	Course policies and organization Sets Racket Logic	Syllabus BP, Chapter 1[1] Slides BP, Chapter 2	 Quiz1 Assn1 Lab 1
2	6/26 - 6/29	Logic Proof Proof	BP, Chapter 2 BP, Chapter 4, 5 BP, Chapter 6, 7	Quiz2 Assn2 Lab 2
3	7/3, 7/5-6	July 4th Holiday Racket Proof Proof	 Slides BP, Chapter 8, 9 BP, Chapter 10	 Quiz3 Lab 3
4	7/10 - 13	Relations Functions Cardinality of Sets	BP, Chapter 11 BP, Chapter 12 BP, Chapter 13	Assn3 Quiz4 Lab 4
5	7/17 - 20	Finite Automata and Regular Languages	TC, Chapter 2	Quiz5 Assn4 Lab 5
6	7/24 - 27	Context Free Languages Final Review	TC, Chapter 3.1-3.4	Quiz 6 Final Exam Lab 6

[1] BP stands for *Book of Proof*, TC stands for *Theory of Computation*