CptS 355 - Programming Language Design

Homepage:
The home page for this course is hosted at:
 http://www.eecs.wsu.edu/~arslanay/CptS355/
The course announcements, lecture notes, recorded lecture videos and announcements will be posted on **Blackboard**.

Class Meeting Times and Location:
Mon, Wed, Fri 9:10am- 10:00am, CLEV 30W (Aug21—Dec8)

Instructor:
Name: Sakire Arslan Ay
Office: EME B57
Telephone: (509)335-4089
E-mail: sakire.arslanay@wsu.edu
Office Hours: Mon, Wed, Fri: 1:10pm-2:50pm

Teaching Assistants:
Name: Yongjun Chen
Office: Sloan312 (only during office hours)
E-mail: yongjun.chen@wsu.edu
Office Hours: Tue: 2:00pm-3:30pm Wed: 10:00am-11:00am

Name: Mark Holy
Office: TBA
E-mail: TBA
Office Hours: TBA

Text Book:
We will not follow any textbooks closely. For SML, you may use the following as a reference.
-   Programming in Standard ML, Robert Harper, Carnegie Mellon University, Spring 2011
    (http://www.cs.cmu.edu/~rwh/isml/book.pdf)
-   Practical Foundations for Programming Languages, Robert Harper, Cambridge University Press, 2012.
    (ISBN-10: 1107029570)
Appendix B includes links to other resources for ML and Racket languages.

Course Overview:
This 3 credit course provides an introduction to basic concepts in the design of programming languages. Students are expected to already be familiar with the C /C++ or Java programming languages. Examples of concepts will be presented using a variety of languages (ML, Racket, Postscript, Java, and Python, for instance). At the end of the course students will be familiar with the concepts that are commonly available in widely used programming languages and have experience using them in several contexts.

Prerequisites: "Cpts 223 Advanced Data Structures"; certified major in CptS, CptE, or EE.

Course Learning Objectives:
This course will help students to achieve the following objectives.
- ✓ Understand components of programming languages including control structures, names, types, objects, exceptions, etc.
- ✓ Understand different kinds of programming language such as imperative, functional and logic languages.

- ✓ Demonstrate skill in using several programming languages.
- ✓ Learn how syntax and semantics are formally specified.
- ✓ Develop a basic understanding of programming language implementation, especially insofar as the implementation impacts the design.
- ✓ Develop the skills necessary to learn new programming languages quickly

In the context of the CptS degree programs' objectives, this course contributes to achievement of the outcomes:

- ✓ An ability to apply knowledge of computing and mathematics appropriate to the discipline. The indicator of achievement of this outcome relevant for this course is: design algorithms and data structures to solve specific problems.
- ✓ An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs. Indicators of achievement of this outcome relevant for this course include: recognizing when theory dictates that a solution is impractical or impossible; generating multiple design concept solutions for a problem; evaluating design concepts and choosing the most promising; implementing the selected approach to obtain a solution; validating a solution and identifying reasons for differences between actual and expected results.
- ✓ A recognition of the need for, and an ability to engage in, continuing professional development. Indicators of achievement of this outcome relevant for this course include: seeking and evaluating outside (of the classroom) sources; solving problems using unfamiliar software packages, languages or systems; recognizing gaps in personal knowledge and identifying what additional knowledge is needed and methods for obtaining that knowledge; making reference to previous learning and applying it in novel situations.
- ✓ An ability to use current techniques, skills, and tools necessary for computing practice. Indicators of achievement of this outcome relevant for this course include: using current programming languages, operating systems, and software libraries to implement solutions; using current tools to manage data and processes related to development of solutions.

Outline:

Topics	# of Lectures
Introduction ; Overview of programming languages	2
Partial functions and computability	2
Functional programming	1
ML(Variable Bindings, Functions, Tuples, Lists, Options, Pattern Matching, Tail Recursion, High-order functions, Data types , Recursive Types, Trees)	6
Types in programming : type systems and inference	2
Racket (List Processing, Higher-order functions, Thunks, Laziness)	4
Python Basics	
Scope and scoping	2
Postscript (Dynamic Scoping)	2
Parameter Passing, Function Closures	2
Concepts in Object Oriented Languages: Java	2
Java classes and inheritance; Java systems architecture	1
C++: Objects and multiple inheritance	1

Attendance and Course Requirements:

Attendance is expected at all lectures. Lecture notes will be posted on Blackboard, but reading the notes is not an adequate substitute for attendance. You are also expected to participate in class discussions to a reasonable extent. This aids learning and provides valuable feedback on the lecture.

We will be using a "classroom response system" (CRS) that allows me to get responses from all students present instead of just a vocal few. You need to bring your laptop or web-capable phone or tablet to each class session to use the system. You are expected to respond thoughtfully each time. Markedly deficient response rates may negatively affect your grade. The CRS will also be used to take attendance each day.

5% of your course grade will be based on your attendance. If you attended 70% of the lectures, your will get full attendance credit. Above 70% will be considered for extra credit. If you attended less than 70%, your attendance grade will be calculated based on the percentage of lectures you attended.

I will record lectures (audio + videos of the computer screen) and make them available on Blackboard. Technical difficulties sometimes arise and occasionally, so don't count on these!

Attendance and assignment/project submissions on time is a strong indication that you care about this class and you put effort to succeed. The instructor will look at your attendance scores and assignment/project submission dates before finalizing letter grades.

Assignments, Quizzes, and Tests:

There will be two midterm exams and one final exam. The midterm exams will be given in class. All exams are closed book and notes. However, you are allowed to bring a one-page notes sheet to the exams (you may use both sides of the sheet).
There will be 6 programming projects (around every 10 to 15 days) which will be posted on Blackboard. Turn-in will be via file upload (onto Blackboard) and you may need to demo certain projects to the course TA. The submission instructions for the assignments will be provided along with the assignment descriptions. The tentative deadlines for the assignment are listed on the course schedule on Blackboard. Please make sure to check the announcements regularly for updates.

The following allocation of grade percentages is tentative and weights are subject to change as circumstances dictate.
Overall Grading:
1. Midterm-1.. 13%
2. Midterm-2.. 13%
3. Final.. 19%
4. Quizzes.. 5%   (average of the 2 quizzes)
5. Programming assignments......... 45%   (7 projects : 5%, 6%, 6%, 5%, 12%, 5%, 6%)
6. Attendance 5%

Letter grades will be assigned based on the scale shown below. **The scale ranges will be adjusted according to the average and spread (standard deviation) of the student scores**. The below scale assumes class average is between 76% and 80%.

Letter Grade	A	A-	B+	B	B-	C+	C	C-	D+	D	F
Total Score	93% - 100%	90% - 93%	86% - 90%	83% - 86%	80% - 83%	76% - 80%	73% - 76%	70% - 73%	66% - 70%	60% - 66%	0% - 60%

CptS students are reminded that a course grade of C or better is required in order for this class to be counted in a computer science degree program. To achieve a grade of C or better in the course, you must separately achieve an average grade of 60% or better on the exams and an average grade of 70% or better on the programming projects.

Makeup exams: If you will miss an exam due to an official WSU activity or other foreseeable reason you must let me know at least 7 days in advance to schedule a makeup. If you are sick on the day of an exam contact me by email or phone as soon as possible (before the exam).

Quizzes: Quizzes are closed book and notes. Your quiz grade will be the average of your two quizzes. Makeups for quizzes will be given within 3 days after the quiz date.
Each student may request a make-up exam or quiz once throughout the semester.

Late Work Policy:

Late work will be penalized 5% for each day or part of a day that the assignment is turned in late. No assignment will be accepted more than 5 days late.

The instructor may make exceptions in the above policy for documented and viable excuses. Students should inform the instructor about their situation prior to the deadline and document their excuse. **Each student may request a single exception during the semester.**

Re-grading Policy:

All homework and projects will be marked as soon as possible after the due date and returned during lecture OR by email. If you discover that a mistake was made in grading, the work may be submitted for re-grading no later than one week after the assignment is returned. This is the only window of opportunity for re-grading. In order to request re-grading, write or type a detailed explanation of the reasons for the re-grading. Submit the explanation along with what was originally turned in to the teaching assistant or professor. Do not orally ask either the professor or TA to re-grade any piece work.

Academic Integrity Policy:

Learning difficult or complicated material is often facilitated by discussions between students and I encourage you to form study groups to try to solve difficult problems together.

However, work you submit must be your own. Exams, quizzes, homeworks, programming projects, and attendance are subject to the WSU academic integrity policy. **Do not cheat in any way: do your own work!** Doing your own work means that you must turn in your own, original work. I define CHEATING to be any attempt to avoid any required obligation for the course, i.e.,:

- Sharing solutions during exams or using any additional material in addition to the allowed (a sheet) notes sheet.
- Sharing solutions or code with your classmates or copying code from solutions/programs of prior semesters' students. It means you do not turn part of a solution you found on the web, nor do you copy code from an assignment solution (you obtained from a friend) from prior semesters. We will process all programming assignment submissions using Stanford's free plagiarism detection software called MOSS. In programming projects, working together may extend to figuring out overall strategies for solution but you may not work together to write the actual code that you submit.
- Having your friend mark you as present in class or signing in as present in class when you are actually elsewhere.

Students who violate these rules or WSU's Standards of Conduct for Students (WAC 504-26-010 (3)) **will fail the assignment** (receive 0 for that assignment or exam) and **be reported to the Office of Student Standards** and Accountability. **A second violation will result in failing the course without having the option to withdraw.** A pattern of academic integrity violations may result in your being decertified from the major. Note that penalties for cheating apply to both a person who turns in copied work and to a person who allows their work to be copied. WSU definitions and procedures for cases of academic dishonesty are given at the URL: https://deanofstudents.wsu.edu/default.asp?PageID=4295

Students with Disabilities:

Reasonable accommodations are available for students with a documented disability. If you have a disability and need accommodations to fully participate in this class, please either visit or call the Access Center (Washington Building 217; 509-335-3417) to schedule an appointment with an Access Advisor. All accommodations MUST be approved through the Access Center. Additional information can be viewed at the URL http://drc.wsu.edu

Campus Safety:

The Campus Safety Plan, which can be found at http://safetyplan.wsu.edu, contains a comprehensive listing of university policies, procedures, statistics, and information relating to campus safety, emergency management, and the health and welfare of the campus community. The left side bar at this safety plan homepage contains many important links to safety information.

Stress Management:

Students sometimes encounter stress and anxiety issues associated with homework, tests, or life situations that can be an obstacle to success in their courses. WSU Counseling Services (335-4511) offers information including a test anxiety workshop conducted several times each semester. Visit http://counsel.wsu.edu/testanxiety. http://counsel.wsu.edu/relaxationroom/ has more information about general stress issues.

Appendix – A

Other Policies and Instructor's Expectations:

- I expect you to demonstrate critical thinking across the spectrum of course work.
- I expect you to engage in active learning: speak up when you don't understand, question assumptions, relate course material to your experience outside class, seek out additional experience and reading related to the class.
- I expect you to promptly review feedback you receive from me, the TA, or other students; to actively clarify the feedback if the material is still unclear; and to incorporate the feedback in your future work.
- I expect you to spend adequate time on the course. Adequate time includes getting enough rest so that time you spend on course tasks is well-spent time. Adequate time includes proofreading and reviewing your assignments before you hand them in.
- I expect you to have high expectations of yourself: set goals for yourself and try to do your very best. Consciously think about the balance between what you do to earn a grade and what you do to learn. (If I'm doing something that puts these in opposition to each other please let me know.)

Instructor's Advice:
- Don't wait until the last minute to do homework or projects. Computers break down, people get sick, etc. These are not sufficient excuses for an extension, especially given the automatic 5 grace days.
- Save early; save often; backup frequently! Sometime you may want to go back to an earlier version if you have an error you can't resolve!
- Come see me or the TA if you are confused. Don't wait for office hours -- send a question by e-mail.
- Come see me and the TA anyway. We'd like to know more about you.

Appendix – B

Links to Online Books and Resources:

SML resources
1. http://www.smlnj.org (links to many resources)
2. Standard ML of New Jersey User's Guide
3. The Standard ML Basis Library
4. SML/NJ Literature : tutorials, books, and documentation

Racket Sources:
1. The Racket Guide
2. Racket Documentation   and   Racket Resources