Instructor: Dr. Mircea Nicolescu
E-mail: mircea@cse.unr.edu
Phone: (775) 784-4356
Office: SEM 232
Office hours: Thursday: 10:30am-1:30pm

Class webpage: http://www.cse.unr.edu/~mircea/Teaching/cs326/

Teaching Assistant: TBA
E-mail: TBA
Office: TBA
Office hours: TBA

Lectures:
Tuesday, Thursday: 9am-10:15am, SEM 234

Labs:
none

Important Notes and Dates:
- Midterm Exam: October 18
- Final Exam: TBA
- Homework assignments: about every 3 weeks

Required Textbook:

Supplemental Books:
- none

Software:
- Scheme
- Prolog
- Java

Course Description:
Catalog:
An overview of programming languages; features, structures, and implementation; examples taken from various programming paradigms. Introduction to formal specifications of languages.
Prerequisites:

Courses:  
- CS 302 (Data Structures) with a “C” or better.

Topics:  
- Good knowledge of at least one programming language (such as C++)  
- Familiarity with data structures (such as linked lists and trees)

Course Objective:

Students will demonstrate an understanding of the fundamental principles underlying programming languages and demonstrate an ability to design and implement applications in a variety of language paradigms.

Student Outcomes and Course Outcomes:

The course outcomes are skills and abilities students should have acquired by the end of the course. These outcomes determine how the general CSE Student Outcomes apply specifically to this course. All CSE Student Outcomes are listed in the next subsection and those relevant to this course are identified in the following Table.

<table>
<thead>
<tr>
<th>CSE Student Outcomes</th>
<th>Course Outcomes</th>
<th>Assessment Methods/Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Students are capable to understand and apply formal methods in language specification and design (context-free grammars, regular expressions, parse trees, scanning, parsing). Students demonstrate a thorough understanding of the fundamental issues in language implementation (naming, control flow, data types, subroutines), and make more effective use of languages they already know.</td>
<td>Formally define simple languages or subsets of existing programming languages, and show how scanners/parsers work with such language definitions. Explain how a given program works under a given implementation, identify appropriate implementation solutions for given situations, propose alternative or improved implementation solutions.</td>
</tr>
<tr>
<td>2</td>
<td>Students are able to make use of the fundamental concepts learned in this course for improving their programming implementation skills.</td>
<td>Write programs in different programming languages, illustrating various implementation aspects.</td>
</tr>
</tbody>
</table>
Students are able to understand how the historical evolution of programming languages has been driven by the tradeoffs involved in language design and implementation, as illustrated by case studies from various programming languages.

Compare different implementation approaches that have been adopted in various languages.

CSE Student Outcomes:
1. Identify, formulate, analyze, and solve complex computing or engineering problems by applying principles of computing, engineering, science, and mathematics.
2. Design, implement, and evaluate a computing or engineering solution to meet a given set of requirements, with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.
3. Communicate effectively in a variety of professional contexts, with a range of audiences.
4. Recognize professional responsibilities and make informed judgments in engineering and computing practice based on legal and ethical principles, considering the impact of solutions in global, economic, environmental, and societal contexts.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline, creating a collaborative and inclusive environment, establishing goals, planning tasks, and meeting objectives.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.
7. Develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.
8. Acquire and apply new knowledge as needed, using appropriate learning strategies.

Course Topics:
- Introduction – The Study of Programming Languages
- Programming Language Syntax
- Names, Scopes and Bindings
- Control Flow
- Data Types
- Subroutines and Control Abstraction
- Data Abstraction and Object Orientation
- Functional and Logic Languages
- Concurrency

Course Policies:
- Students are expected to attend, and be on time, for every class. This demonstrates professionalism and consideration for your fellow students and your instructor. While the course does not have an attendance policy, students who miss class and/or are late for class may experience an impact on their grade by missing classroom activities.
- Students are expected to turn in all assigned materials in a timely manner.
• Students are expected to demonstrate professionalism and courtesy by either silencing or turning off all cell phones and/or other alarm or audible indicator devices.

• The instructor reserves the right to add to, and/or modify any of the above policies as needed to maintain an appropriate and effective educational atmosphere in the classroom. In the case that this occurs, all students will be notified in advance of the implementation of the new and/or modified policy.

UNR Athletics:
• If you are involved with any university-sponsored athletic activities that will have an impact on your attendance, please provide your instructor with a letter from your coach and/or the UNR Athletic Department as soon as possible, but no later than the end of the second week of classes. This should include the official schedule of your activities which will impact your attendance throughout the semester.

Assignments, Examinations and Grading:

Homework Assignments:
• There will be a number of homework assignments, some of which will include programming. The homework assignments and their due dates will be posted on the course web page. Homework assignments are due on the specified date at the beginning of the class.

Late Submission Policy:
• No late assignments will be accepted.

Exams:
• There will be one midterm exam and one final exam. Both exams will be closed books, closed notes. Permissions to take exams on other dates than scheduled will not be given, except for extreme medical emergencies.

Grading Structure:
• The final score will be computed as follows:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework assignments</td>
<td>40%</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>25%</td>
</tr>
<tr>
<td>Final exam</td>
<td>30%</td>
</tr>
<tr>
<td>Attendance and class participation</td>
<td>5%</td>
</tr>
</tbody>
</table>

Letter Grades:
• The letter grade will be computed according to the following table. Some upward adjustment may occur, but do not count on it.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Letter</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>88-100</td>
<td>A-, A</td>
<td>40%</td>
</tr>
<tr>
<td>75-87</td>
<td>B-, B, B+</td>
<td>25%</td>
</tr>
<tr>
<td>62-74</td>
<td>C-, C, C+</td>
<td>30%</td>
</tr>
<tr>
<td>50-61</td>
<td>D-, D, D+</td>
<td>5%</td>
</tr>
<tr>
<td>&lt; 50</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>
Academic Dishonesty:
Cheating, plagiarism or otherwise obtaining grades under false pretenses constitute academic dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include filing a final grade of "F"; reducing the student's final course grade one or two full grade points; awarding a failing mark on the coursework in question; or requiring the student to retake or resubmit the coursework. For more details, see the University of Nevada, Reno General Catalog.

Disability Services:
Any student with a disability needing academic adjustments or accommodations is requested to speak with the Disability Resource Center as soon as possible to arrange for appropriate accommodations.

Academic Success Services:
Your student fees cover usage of the University Math Center (775) 784-4433, University Tutoring Center (775) 784-6801, and University Writing Center (775) 784-6030. These centers support your classroom learning; it is your responsibility to take advantage of their services. Keep in mind that seeking help outside of class is the sign of a responsible and successful student.

Audio and Video Recording:
Surreptitious or covert video-taping of class or unauthorized audio recording of class is prohibited by law and by Board of Regents policy. This class may be videotaped or audio recorded only with the written permission of the instructor. In order to accommodate students with disabilities, some students may be given permission to record class lectures and discussions. Therefore, students should understand that their comments during class may be recorded.

University Math Center:
The University Math Center (UMC) is focused on helping students with mathematical and statistical concepts. While mathematics is used extensively in engineering, the UMC does not have the resources to help students with engineering courses. Engineering students are encouraged to use the UMC for help in their math classes, and they are welcome to use its computer lab and study area any time – regardless of course. However, UMC tutors cannot answer questions regarding engineering courses.