CSC301: Fundamentals of Programming Languages

Syllabus – Fall 2020

Time: Section 1 MWF 10-10:50, Location: Online

Webpage: homepage.cs.uri.edu/faculty/hamel/courses/home/csc301
Prerequisites: CSC212

Instructor:
Prof. Lutz Hamel
email: lutzhamel@uri.edu
office: online meetings only

Course Description
Language enables thought. In this course we study a class of formal languages known as programming languages. Similar to natural languages, these formal languages enable us to reason about algorithms and procedures to solve computational problems on computers. However, their formal nature restricts the kind of meanings particular language constructs can assume and therefore makes them amenable for the execution on a computer.

Over the years many different programming language dialects have evolved to address particular technical issues, e.g. object-oriented languages, real-time languages, database query languages, logic languages, etc. Here we study the major structures of modern programming languages. Understanding not only the syntax of a language but also the semantics and implementation techniques of this language will allow you to design better programs. Having deeper insights into the design of a programming language will also enable you to learn new programming languages much faster. Having a thorough understanding of today's languages allows you to design the programming languages of tomorrow.

Objective
Upon completion of this course

- You will be able to discern and contrast the major programming language paradigms in use today.
- You will be able to pick an appropriate language for the job at hand.
- You will have deeper insight into the evolution of programming languages.

Text
Class notes.
Software
Throughout this course we will be using various programming language and software development environments including: Rust, Haskell, and Prolog. More details will be given on the course website.

Grading
Quizzes and Labs 40%
Midterm 30%
Final 30%

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Start %</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>50</td>
</tr>
<tr>
<td>D+</td>
<td>55</td>
</tr>
<tr>
<td>C-</td>
<td>60</td>
</tr>
<tr>
<td>C</td>
<td>65</td>
</tr>
<tr>
<td>C+</td>
<td>70</td>
</tr>
<tr>
<td>B-</td>
<td>75</td>
</tr>
<tr>
<td>B</td>
<td>80</td>
</tr>
<tr>
<td>B+</td>
<td>85</td>
</tr>
<tr>
<td>A-</td>
<td>90</td>
</tr>
<tr>
<td>A</td>
<td>95</td>
</tr>
</tbody>
</table>

Policies
• Check the website (often)! I will try to keep the website as up-to-date as possible.
• Class attendance, promptness, participation, and adequate preparation for each class are expected. If you are absent, it is your responsibility to find out what you missed (e.g. handouts, announcements, assignments, new material, etc.)
• Make-up quizzes and exams will not be given without a valid excuse, such as illness. If you are unable to complete a scheduled examination due to valid reasons, please inform myself, or the department office in Tyler Hall, prior to the exam time. Under such circumstances, you are not to discuss the exam with any other class member until after a make-up exam has been completed.
• All work is to be the result of your own individual efforts unless explicitly stated otherwise. Plagiarism, unauthorized cooperation or any form of cheating will
be brought to the attention of the Dean for disciplinary action. See the appropriate sections (8.27) of the University Manual.

- **Software piracy** will be dealt with exactly like stealing of university or departmental property. Any abuse of computer or software equipment will subject to disciplinary action.

- Any student with a documented disability is welcome to contact me as early in the semester as possible so that we may arrange reasonable accommodations. As part of this process, please be in touch with Disability Services for Students Office at 302 Memorial Union, Phone 401-874-2098.

**Special Covid-19 related statements:**

The University is committed to delivering its educational mission while protecting the health and safety of our students. At this uncertain time, those concerns include minimizing the potential spread of COVID-19 within our community. While the university has worked this summer to create a healthy learning environment for all, it is up to all of us to ensure our campus stays that way.

As members of the URI community, students are required to comply with standards of conduct and take precautions to keep themselves and others safe. Students are required to comply with Rhode Island state laws, including the Rhode Island Executive Orders related to health and safety, ordinances, regulations, and guidance adopted by the University as it relates to public health crises, such as COVID-19.

An addendum on policies and guidelines concerning your obligations during this crisis has recently been integrated into the Student Handbook. These obligations include:

- Wearing of face masks by all community members when on a URI campus in the presence of others
- Maintaining physical distancing of at least six feet at all times
- Following state rules on the number of individuals allowed in a group gathering
- Completing a daily health self-assessment also available through the Rhody Connect app before coming to campus
- Submitting to COVID-19 testing as the University monitors the health of our community
- Following the University’s quarantine and isolation requirements

If you answer yes to any of the questions on the daily health assessment, do not come to class. YOU MUST STAY HOME/IN YOUR ROOM and notify URI Health Services via phone at 401-874-2246 immediately.

If you are already on campus and start to feel ill, you need to remove yourself from the public and notify URI Health Services via phone immediately at 401-874-2246 and go home/back to your room and self-isolate while you await direction from Health Services.

If you are unable to attend class, please let me know and we will work together to ensure that course instruction and work is completed for the semester.

**Tentative Schedule**

1. Why study programming languages?
2. Meet our languages:
   * Rust - OO/Imperative
   * Haskell - Functional
   * Prolog - Logic

3. Rust
   * The basics
   * Pattern matching oriented programming
   * Types
     * Type checking
     * static vs dynamic
     * Type hierarchies
     * Generics
   * Objects and traits
   * Higher order functions
   * Iterators (internal/external/generator)

4. Interlude - Language specification
   * Syntax/Semantics
   * Grammars
   * Parse tree/derivation trees
   * Semantics: interpretations of the parse tree
   * Ambiguous grammars
   * Lexical analysis
   * Parser generators
     * Lex/YACC

5. Interlude - Language Tools
   * Tool chains
   * Interpreters vs compilers
   * Debuggers

6. Interlude - Memory management
   * Typical process memory
   * Parameter passing
     * Pass by value
     * Pass by reference

7. Haskell
   * Lists and tuples
   * Types and type classes
   * Functions and pattern matching
     * Let, where, and case
   * Recursion/thinking recursively
   * Higher order functions (mapping)
   * Algebraic data types
     * Functional IO

8. Interlude - Recursion
* How do programming language implement recursive functions?
  * The runtime stack and stack frames
  * Using a debugger to look at the runtime stack

9. Interlude - Polymorphism
  * Duck typing
  * Subtype polymorphism
  * Parametric polymorphism

10. Prolog
  * First-order logic
  * Predicates and quantification
  * Modus ponens
  * Logic programs
  * Facts and rules
  * Querying/executing a program
  * Proof trees
  * Arithmetic
  * IO

11. Interlude - Scoping
  * Scoping rules
  * Name spaces
  * Static vs. dynamic scoping

12. Interlude - Formal semantics
  * Semantics via abstract interpretation
  * Operational semantics
  * Building abstract interpreters using Prolog

13. Semester Review