University of Rhode Island

Department of Computer Science and Statistics
CSC402, Programming Language Implementation, Section 1, Fall 2017

Instructor:	Lutz Hamel
Office Location:	Tyler Hall 251
Telephone:	401-480-9499
Email:	hamel@cs.uri.edu
Class Days/Time:	MWF 9-9:50am
Classroom:	Bliss Hall Rm 304
Prerequisites:	CSC301
Webpage:	homepage.cs.uri.edu/faculty/hamel/courses/2017/fall2017/csc402

Course Description

- Have you ever wondered how the syntax highlighter in Eclipse works?
- Have you ever wondered how languages like PHP and HTML are implemented?
- How about Python and Pearl?
- Or for that matter, Java and JavaScript?
- What is the difference between interpreting a programming language and translating/compiling it?
- What is the difference between an interpreter and a virtual machine?

If any of these questions interest you then CSC402 is for you. We will spend the semester looking at programming language implementations: from syntax highlighters to code analyzers, from interpreters and virtual machines to compilers.

As part of the course we will construct interpreters and translators for domain specific languages such as calculator languages and command line languages for steering your favorite game character. The course will also include one large semester project of a language implementation project of your choosing. This could be a graphics language, a new programming language (think Ruby), a domain specific language such as PHP or a new command line shell interpreter for Unix/DOS.

Course Goals
The goal of the course is to give you a solid foundation with respect to programming language implementation that includes:

- grammar construction
- parsing techniques
- intermediate representations
- abstract syntax tree construction
- tree pattern matching techniques

We will study a number of different programming language implementation techniques including,

- compilers
- interpreters, and
- virtual machines.

These tools will enable you to add domain specific and general programming language implementations to your tool chest to solve difficult engineering problems.

Upon successful completion of this course, each student will be able to:

- Understand the difference between compilers and interpreters.
- Use grammar specification tools effectively.
- Design and implement domain specific and general purpose programming languages.

NOTE: Students taking this course for graduate credit (CSC502) will have to complete an additional Unix/Linux based project.

Required Texts/Readings

Textbook

none – free online text book available

Classroom Protocol

- 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.)
- **Late assignments** will **not** be accepted.
- **Make-up quizzes** and **exams** will **not** be given without a valid excuse, such as illness. If you are unable to attend 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 handled according to the University Man- ual section 8.27.10 through 8.27.21 (see www.uri.edu/facsen/8.20-8.27.html). The penalty for cheating or plagiarism can range from a zero score on the assignment to a failing grade for the course.
- **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.

Exams, Assignments, and Grading Policy

Course Grade Composition:	
Homework and Programming Assignments	40%
Midterm	30%
Final	30%

Grading Scale:

95 - 100 A
90 - 94.9 A-
85 - 89.9 B+
80 - 84.9 B
75 - 79.9 B-
70 - 74.9 C+
65 - 69.9 C
60 - 64.9 C-
55 - 59.9 D+
50 - 54.9 D
0 - 49.9 F

Homework consists of exercises to familiarize you with common tools and concepts in programming language implementation. Programming assignments are typically projects that can be completed within a couple of days. The midterm and the final comprise major projects and you should budget your time accordingly. Homework and programming assignments are assigned on a weekly basis.

Disability Accommodations and Opportunities

Any student with a documented disability should contact me early in the semester so that we can make reasonable accommodations to support your success in this course. You should also contact Disability Services for Students, Office of Student Life, 330 Memorial Union, 874-2098

Tentative Course Schedule

Week 1	Programming Languages and their Processors
Week 2	Language tools
Week 3	Program Analysis
Week 4	Tree Walking
Week 5	Optimizing Compilers
Week 6	Scope & Symbol Tables

Week 7	Functions
Week 8	Type systems
Week 9	Structured data types
Week 10	Higher-Order Programming
Week 11	Compiling for Real Machines