CSC402: Programming Language Implementation

Syllabus – Fall 2013

Time: Section 1 9-9:50, Location: Tyler Hall Rm 109
Webpage: http://homepage.cs.uri.edu/faculty/hamel/courses/2013/fall2013/csc402
Prerequisites: CSC301

Instructor:
Prof. Lutz Hamel
email: hamel@cs.uri.edu
office: Tyler Hall

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.

The goal of the course is to give you a solid foundation with respect to programming language implementation which includes grammar construction, parsing techniques, intermediate representations, tree construction, tree pattern matching techniques, among many other topics. 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.

Required Texts

The Definitive Antlr Reference: Building Domain-Specific Languages. Terrence Parr, Pragmatic Bookshelf, 2007 – be aware we use ANTLR Version 3. The best way to get this book is as a download from this website: http://pragprog.com/book/tpantlr/the-definitive-antlr-reference

Software

The main software we will be using in this course is ANTLR Version 3 and Java. Both of these are available as public domain software. More details on the course website.

Grading

Homework, Quizzes, and Programming Assignments	40%
Midterm	30%
Final	30%

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.)
- **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 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.

Tentative 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