CSC301: Fundamentals of Programming Languages

Syllabus – Fall 2014

Time: Section 1 MWF 10-10:50, Location: Crawford Hall Rm 222
Webpage: http://homepage.cs.uri.edu/faculty/hamel/courses/2014/fall2014/csc301
Prerequisites: CSC212

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

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

Modern Programming Languages: A Practical Introduction, Adam Brooks Webber, Franklin, Beedle & Assoc., any edition.

Software

Throughout this course we will be using various programming language and software development environments including: ML, Java, and Prolog. More details will be given on the 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 be accepted with a penalty of 5% per day late until I pass the homework back. Once I pass back the homework no late assignments will 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

Chapter 1: Programming Languages
Chapter 2: Defining Program Syntax

Week 2

Chapter 3: Where Syntax Meets Semantics
Chapter 4: Language Systems

Week 3

Chapter 5: A First Look At ML
Chapter 6: Types

Week 4

Chapter 7: A Second Look At ML
Chapter 8: Polymorphism

Week 5

Chapter 9: A Third Look At ML
Chapter 10: Scope

Week 6

Chapter 12: Memory Locations For Variables
Chapter 13: A First Look At Java
Chapter 14: Memory Management
Chapter 15: A Second Look At Java

Week 7

Chapter 16: Object Orientation
Chapter 18: Parameters

Week 8

Chapter 19: A First Look At Prolog

Week 9

Chapter 20: A Second Look At Prolog
Chapter 22: A Third Look At Prolog

Week 10

Chapter 23: Formal Semantics

Week 11

Chapter 24: The History of Programming Languages