Compiler Construction
CSCI-742
Term 20135
Handout 1
January 28, 2014

Syllabus

General Information

Instructor: Matthew Fluet
E-mail: mtf@cs.rit.edu
Office hours: Tu 11:00am – 12:00pm; GOL-3555
We 10:00am – 12:00pm; GOL-3555
Th 11:00am – 12:00pm; GOL-3555
or by appointment

Lectures: Section 01 TuTh 1:00pm – 2:15pm; GOL-2690

Website: http://www.cs.rit.edu/~mtf/teaching/20135/cc
http://mycourses.rit.edu

FusionForge Server: https://asgard.cs.rit.edu:443

Course Description

This course discusses design and implementation of language processors and translators. Topics include lexical, syntactic, and semantic descriptions, algorithms for analysis tools, and programming techniques, as well as interpreters and code generation for typical computer architectures.

Prerequisites

- CSCI-603 (Advanced C++ and Program Design) and CSCI-605 (Advanced Java Programming) (with B or better in both courses) and CSCI-661 (Foundations of Computer Science Theory)
- or CSCI-141 (Computer Science I - Introduction to Computational Problem Solving) and CSCI-142 (Computer Science II - Computational Problem Solving with Software Structures) and CSCI-243 (The Mechanics of Programming) and CSCI-661 (Foundations of Computer Science Theory)
- or permission of instructor
Course Goals

The principles of compiler construction are fundamental to the skill set of a computer scientist. This course focuses on descriptive languages, data structures, and algorithms used to implement programming-language processors and implementation tools. Topics include lexical, syntactic, and semantic descriptions, analysis tools, and programming techniques, as well as interpreters and code generation for typical computer architectures.

Course and Program Outcomes

Course learning outcomes:

- The student will be able to explain the principles of interpreter, compiler, and runtime architecture and implementation.
  Program outcome(s): 2
  Evaluation: exams

- The student will be able to design a simplified programming language and implement an interpreter or pseudo-code compiler using appropriate tools.
  Program outcome(s): 2
  Evaluation: programming project

- The student will be able to explain different techniques and generator algorithms for lexical and syntax analysis of context-free languages.
  Program outcome(s): 2
  Evaluation: exams

- The student will be able to explain and implement type checking.
  Program outcome(s): 2
  Evaluation: exams, programming project

- The student will be able to explain and implement the principles of scoping, parameter passing, code generation and runtime memory management.
  Program outcome(s): 2, 3
  Evaluation: exams, programming project

Program Outcomes:

- (CS Graduate Program Outcome 2) Demonstrate a depth of knowledge in a selected area in the discipline.

- (CS Graduate Program Outcome 3) Communicate effectively in a professional environment.

Grades

Grades will be assigned based on the following grading scheme:

- Attendance & Participation: 5.0%
- Homework Assignments: 10.0%
- Programming Projects: 65.0%
- Mid-term Exam: 10.0%
- Final Exam: 10.0%
Important Dates

March 18 (Tue.): Mid-term Exam (in class, 75min)
May 16 (Fri.): Final Exam (10:15am – 12:15pm; GOL-2690)

Text Books

Strongly recommended:
Title: Modern Compiler Implementation in ML
Author: Andrew Appel
Publisher: Cambridge University Press
ISBN: 978-0521607643

Title: Compilers: Principles, Techniques, & Tools (Second Edition)
Author: Alfred Aho, Monica Lam, Ravi Sethi, Jeffrey Ullman
Publisher: Prentice Hall
ISBN: 978-0321486813

Recommended:
Title: ML for the Working Programmer
Author: Lawrence Paulson
Publisher: Cambridge University Press
ISBN: 978-0521565431

Optional:
Title: Elements of ML Programming
Author: Jeffrey Ullman
Publisher: Prentice Hall
ISBN: 978-0137903870

Title: The Standard ML Basis Library
Editors: Emden Gasner and John Reppy
Publisher: Cambridge University Press
ISBN: 978-0521794787

Course Policies

Attendance & Participation

Students are required to attend and expected to participate in class. Participation means being an engaged student: asking and answering questions, not simply attending class.

The use of cell phones and audio players is prohibited during class. If you must take a phone call, please leave the classroom immediately and do not return until you have ended the phone call.
The use of a laptop (or notebook or netbook) computer is permitted during class only for the purpose of taking notes. Persistent use of a laptop for other activities will result in 0 credit for your *Attendance \\& Participation* grade.

Assigned readings should be completed before the lecture section. You are responsible for the material in assigned readings, whether covered during lecture or not.

**Mid-term Exam**

There will be one mid-term exam; see above for the date.

The mid-term exam must be taken at its scheduled time. Make-up mid-term exams will not be administered, unless exceptional circumstances have been discussed with the instructor in advance of the exam date and/or other arrangements have been made.

**Final Exam**

There will be a final exam; see above for the date. The final will be comprehensive and will cover material from the entire course, including readings, lectures, and assignments.

The final exam must be taken at its scheduled time. Any exam conflicts must be reported to the instructor by the end of Week 10 (see the RIT Final Examination Policies).

**Late Policy**

Homework assignments that are submitted in person will generally be due at the beginning of a class period. Project assignments that are submitted electronically will generally be due at 11:59PM on the due date.

Assignments are to be submitted on time. However, to accommodate the occasional difficulty with meeting an assignment due date, each student begins the term with five “extension tokens.” By spending an extension token, you will receive a 24-hour extension on a single assignment. To spend an extension token, you must e-mail the instructor before the assignment is due; you cannot spend an extension token after an assignment’s due date has passed. You may spend at most one extension token on a single assignment. After spending five extension tokens, late assignments will not be accepted.

**Regrading**

After a graded exam or assignment has been returned, you have **one week** to bring any questions about grading to the instructor’s attention. No grade adjustments will be made after this time.
Academic Integrity

As with all courses, the RIT Honor Code RIT Academic Honesty Policy apply. See the Department of Computer Science’s statement on academic integrity for more details.

In this course, all submitted work must be your own work (i.e., written or programmed by you alone, unless explicitly stated otherwise) and must include acknowledgments of any collaborators or sources (other than course text books or handouts) used to produce your submission.

You are encouraged to discuss course material with other students. Discussion of assignments is also allowed, but sharing solutions or code is not allowed.

Common Course Policies

See the Department of Computer Science’s Common Course Policies for more details about rescheduling an exam, course withdrawal, disability services, and academic integrity.

Disclaimer

I reserve the right to make any changes to the syllabus as I deem necessary throughout the course. Minor changes, such as assignment due dates, will be announced orally during class and posted on the course mailing list and home page. Major changes, such as grading percentages, will additionally be provided in writing.