# CSCI-481: Compilers

Spring 2015

## Instructor


| | |
|---|---|
| Instructor: | Brad Richards |
| Lectures: | 481: TH 381 — MWF 11:00–11:50 |
| | 879–3579 |
| Email: | brichards@pugetsound.edu |
| Office: | Thompson 401 |
| Office Hours: | M 2–2:50, Tu 12–12:50, W 2–2:50, F 2–2:50 |
| Tutor Schedule: | See online schedule |


## Contents



## Resources:

- *Modern Compiler Implementation in Java (2e)*, Andrew Appel, Cambridge University Press. (Required text.)
- The MiniJava project web site
- SableCC home page and downloads.
- The MARS MIPS simulator.
- A MIPS instruction reference
- Some useful MIPS information
- Sample Code.
- Other course-related documents.

## Schedule:

| Week | Topic | Reading | Assignment |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1/19 | Intro, Representations, Visitor pattern | Chapter 1 | Visitor |
| 1/26 | Lexical Analysis; Parsing | Chapter 2, 3 | Lexer |
| 2/2 | More parsing... | Chapter 3 | Parser |
| 2/9 | More parsing, Abstract syntax trees | Chapter 4 | |
| 2/16 | Type checking, Semantic analysis | Chapter 5 | AST |
| 2/23 | More analysis | | Symbol Table |
| 3/2 | Activation records / Frames<br>*No Class Wed or Fri* | Chapter 6 | Semantic Analysis |
| 3/9 | Intermediate representation | Chapter 7 | IRT and Variables |
| 3/16 | | Spring Break | |
| 3/23 | More IR: Objects, Arrays | Chapter 7 | IRT for straightline MiniJava |
| 3/30 | More IR: Dynamic Dispatch, Booleans | | |
| 4/6 | Code Generation, Register Allocation | Chapter 8 | Full IRT Programs |
| 4/13 | View Shift Code | Chapter 9 | MIPS Assembly! |
| 4/20 | Wrapping up code generation | | |
| 4/27 | Additional topics / overflow | | |
| 5/4 | Short Week | | |

## Background:

This course gives a formal presentation of programming language translation and compiler writing. It introduces formal languages and automata theory as tools for recognizing high-level computer programming languages, and presents techniques for parsing and translating these languages into intermediate representations and lower-level computer languages that are equivalent to the original input program. Students will each build a compiler for a Java subset as a course project.

This course builds progressively on previously covered material. Therefore, it is essential to attend all classes and keep up with the reading and the assignments. Students are expected to attend all lectures, with exceptions permitted in case of illness and family emergencies, and should do the assigned readings before each class.

The assignments form a crucial part of the course, and students are required to work on them individually. Collaboration with other students is not permitted and will be subject to severe penalties. Please review the Academic Integrity section in The Logger and ask me if you have any questions regarding its application to this course.

Academic accommodations are available for students with disabilities who are registered with the

Office of Disability Services. Please schedule an appointment with the instructor early in the semester to discuss any accommodations for this course which have been approved by the Disability Services Coordinator as indicated in your accommodation letter.

## Grading:

Homework for this class will be submitted electronically, and must be turned in by class time on the due date for full credit (unless the assignment explicitly states otherwise). Late homework will be penalized 5% per day, and will not be accepted more than one week late. The following grade cutoffs are upper bounds — they might come down, but will not be set higher: A = 95, A- = 90, B+ = 88, B = 83, B- = 80, C+ = 77, C = 73, C- = 70, D+ = 67, D = 64, D- = 60, F = <60. Your overall grade is composed as follows:

- 75%: Programming Projects
- 20%: Midterm Exam (March 30th)
- 5%: Course participation

Participation: At the end of the semester you will receive a score of 0 through 5 for participation. The "default" grade will be a 2.5, meaning you were generally physically present and mentally engaged. A higher score will be given to students whose engagement is noteworthy. Examples include: answering and asking relevant questions, noticing mistakes (in a polite and productive manner), active engagement in class activities. A score of lower than 2.5 will be given for students with multiple absences, minor class disruptions, or being mentally absent.

## Project Grade

Completing all of the project checkpoints will produce a working compiler that generates (bad) MIPS assembly code for MiniJava programs. In particular, it will do a naive job of register allocation, and will not be required to do dynamic dispatch when invoking methods. This basic implementation is enough to get you 90% of the project grade, if done well. The following optional extensions can get you some additional points:

| | |
|---|---|
| Dynamic Dispatch | 5% |
| Short-circuited AND | 2% |
| Bounds-checking code for arrays | 2% |
| Initializing arrays with zeros | 1% |
| Better register allocation | Negotiable |
| Additional language features | Negotiable |

## Emergency Response

Please review university emergency preparedness and response procedures posted at www.pugetsound.edu/emergency/. There is a link on the university home page. Familiarize yourself with hall exit doors and the designated gathering area for your class and laboratory buildings.

If building evacuation becomes necessary (e.g. earthquake), meet your instructor at the designated gathering area so she/he can account for your presence. Then wait for further instructions. Do not return to the building or classroom until advised by a university emergency response representative.

If confronted by an act of violence, be prepared to make quick decisions to protect your safety. Flee the area by running away from the source of danger if you can safely do so. If this is not possible, shelter in place by securing classroom or lab doors and windows, closing blinds, and turning off room lights. Lie on the floor out of sight and away from windows and doors. Place cell phones or pagers on vibrate so that you can receive messages quietly. Wait for further instructions.