15 Compilers in 15 Days

Jeremy D. Frens & Andrew Meneely

Calvin College

SIGCSE 2006

Compilers and a January Term

- January Term (a.k.a. “Interim”)
    - Fifteen class days; three weeks.
    - Three hours per day in-class.
    - Only course taken by students.
    - Usually pass/fail.
- Our independant study
    - Fifteen weeks.
    - Four hours per week in-class.
    - Part of normal semester.
    - Taken for a grade.

Ready, set, go!

1 compiler in 10 minutes!

- Input: file with an integer.
- Ouput: file with Parrot code to print the integer.

Agile Toolkit

- Test-driven development
    - Unit testing
    - Refactoring
- Incremental development
    - A complete compiler each day
    - YAGNI

Compiler Topics (Day 1)

- Scanning
  - ANTLR tool
  - Regular expressions
- Code generation
  - Parrot opcodes
- A compiler *front to back*.

Ready, set, go!

Compiler of Day 3

- Input: an arithmetic expression.
- Output: Parrot code which will compute the expression and print the result.

Compiler Topics (Day 3)

- Scanning
  - More interesting regular expressions
- Parsing
  - Infamous expression grammar
  - Why is precedence implemented in the parser?
- More (sophisticated) code generation
  - Need to use stack

What *Could* We Have Seen?

- Peephole optimizations.
- Constant folding.

Other Things in the Paper. . .

- Eclipse good. . .
- CVS good. . .
- Mock objects good. . .

Why?

- TDD is fun!
- Compilers are fun!
- Half the work covers half of the fundamentals.
- Well-balanced view of a compiler.

Current and Future Work

- Other topics
  - 15 Raytracers in 15 Days
  - 15 Interpreters in 15 Weeks (Programming Languages)
- Other technologies
  - Mock objects
  - Acceptance testing (with FitNesse)
  - Pair programming
  - Group work