



Bantam Java Compiler Project

Marc Corliss

Hobart and William Smith Colleges
corliss@hws.edu

E Christopher Lewis

VMware, Inc.
lewis@vmware.com

HOBART
AND
WILLIAM SMITH
COLLEGES

Bantam Java Compiler Project – Marc Corliss

MATHEMATICS
AND
COMPUTER SCIENCE
DEPARTMENT

Compiler Construction

Building a compiler is a valuable learning experience

- Requires understanding language, machine, and everything in between
- Illustrates many areas of computer science (e.g., computational theory, data structures, algorithms)
- Provides challenging software engineering experience

HOBART
AND
WILLIAM SMITH
COLLEGES

Bantam Java Compiler Project – Marc Corliss

MATHEMATICS
AND
COMPUTER SCIENCE
DEPARTMENT

Existing Compilers

Commercial compilers

- + Real language
- Too complex for one semester course

Compilers for simple languages

- + Doable in one semester course
- Language not realistic (e.g., lack OO)

Classroom Object-Oriented Language (COOL) [Aiken, '96]

- + Doable in one semester, OO language, emphasizes type theory
- Language different than existing languages

MiniJava [Appel and Palsberg, '96]

- + Doable in one semester, Java-like language
- Integrated tightly with textbook, OO optional add on, publisher-provided instructor code incomplete

HOBART
AND
WILLIAM SMITH
COLLEGES

Bantam Java Compiler Project – Marc Corliss

MATHEMATICS
AND
COMPUTER SCIENCE
DEPARTMENT

Goals of this Work

Language is subset of existing, well-used language (Java)

Language includes small (yet broad) set of key features

Compiler engineered for simplicity, easy to extend
(for both students and instructors)

Language/compiler thoroughly documented in free manual

Compiler project customizable

HOBART
AND
WILLIAM SMITH
COLLEGES

Bantam Java Compiler Project – Marc Corliss

MATHEMATICS
AND
COMPUTER SCIENCE
DEPARTMENT

Bantam Java

Language and compiler designed specifically for classroom

Language is subset of Java

Language includes some important features of Java
(emphasis was on object-oriented features)

Simple compiler infrastructure, heavily modularized
(can be built in 4 independent projects)

Compiler has some customizable components
(e.g., lexer/parser generators, targets, language*)

Outline of Remainder of Talk

Bantam Java Language

Bantam Java Compiler

Resources

Future Work

Bantam Java Language

Small subset of Java

- Has many key features of Java

Features:

- Object-oriented (supports dynamic dispatch and inheritance)
- Types: int, boolean, Object, String, TextIO, Sys
- Operators: =, +, -, *, /, %, ==, !=, <, >, <=, >=, &&, ||, !, new, object cast
- Control flow: if statements and while loops
- Memory management: allocation via new, deallocation via garbage collector

Excluded from Bantam Java

Arrays

Static members

Primitives besides int and boolean

Modifiers such as public, private, *etc.*

Method overloading

User-defined constructors

Loops besides while (*e.g.*, for), switch statements

(among other things)

Language Summary

Chose to emphasize object-oriented features

(e.g., inheritance, dynamic dispatch, object casting)

- Objects critical component of most modern languages
- Students have less intuition about objects and their implementation than many other language features

Left out "little bang for buck" features

(e.g., modifiers, method overloading, for loops, etc.)

Left out some useful features

(e.g., arrays, static members, floating point types)

- Need compiler to be doable in one semester course
- Will build extended language to include some of these features

Bantam Java Compiler



Written in Java

Split into 4 phases

- Each phase implemented in separate course project
- Use lexer/parser generator to build lexer/parser
- Semantic analyzer and code generator built from scratch

Intermediate representation

- Abstract syntax tree (AST)
- Use visitor design pattern to traverse AST

Customizing the Compiler

Lexer/parser generators

- JLex and Java Cup
- JavaCC

Targets

- Mips/SPIM (simulated)
- x86/Linux (native)

Language*

- Base (already seen)
- Extended (includes arrays, static members, float) (work in progress)

Compiler Summary

Compiler engineered for simplicity and extendibility

- Split into 4 phases, single program representation

Project embodies practical software engineering techniques

- Packaging/modularity, visitor pattern, complex data structures

Compiler has some customizability

- Lexer/parser generators, target machine, language*

Resources

Bantam Java website (<http://www.bantamjava.com>)

Bulletin board

Comprehensive, free, student lab manual

Compiler API available online (generated using javadoc)

Working solution (available to instructors via email)

Future Work

New language features: arrays, static members, float, ...
(currently working on)

Building compiler in Bantam Java itself (w/ Lex Kridler)

New targets: JVM

New course projects: optimization (w/ Lori Pietraszek), ...

Community involvement

Conclusions

Bantam language and compiler designed for classroom

- Designed specifically for course in compilers

Bantam Java Compiler Project has many nice features

- Language subset of Java, includes many key features
- Compiler engineered for simplicity, easy to extend
- Compiler is customizable in several ways
- Project is well documented

Last Words

We hope you try it out and enjoy it!

URL: <http://www.bantamjava.com>

We welcome and encourage feedback on ways for
improvement