

# Teaching with Grace

Kim B. Bruce<sup>1</sup> and Andrew P. Black<sup>2</sup>

- 1 Department of Computer Science, Pomona College  
Claremont, CA 91711, USA  
kim@cs.pomona.edu
- 2 Department of Computer Science, Portland State University  
Portland, OR 97201, USA  
black@cs.pdx.edu

## 1 Grace

Grace [1] is a new object-based language designed for novices. While we are still tweaking some aspects of the language design, and the current implementations are rudimentary, the authors are teaching with Grace this academic year. Kim Bruce used Grace in an introductory course at Pomona College this fall for the first time, and Andrew Black is teaching with it in the spring quarter at Portland State. While the Pomona course was intended primarily to help us find out what aspects of the language and implementation need work, positive student feedback has led us to consider this trial a success. We plan to continue working on the implementations and programming environment over the next year, and to then conduct some careful experiments to compare the Grace version of our introductory courses with the equivalent courses using other object-oriented languages — Java at Pomona College and C++ at PSU.

In a short paper, the best we can do to describe Grace is to list buzz-words: object-oriented, object-based (meaning that objects, not classes, are fundamental), low-overhead, gradually typed, having anonymous functions ( $\lambda$ -expressions) and a straightforward syntax and semantics. A dialect mechanism [3] allows instructors to extend the language for specific assignments (for example, with graphics primitives), to hide features that students have not yet encountered, and to customize error messages. For details, we recommend you visit [www.gracelang.org](http://www.gracelang.org), which includes links to papers on the Grace language design.

## 2 Teaching Gracefully

The implementation of Grace used for the Pomona course was web-based, with program editing, compilation, and execution all taking place in a web browser. This had the advantages that no software had to be installed on students' computers, and that there was little overhead in learning to use the tools. The main weaknesses of this implementation are (1) the gradual and static type systems are incomplete, (2) threads are not available, (although animation is supported using timers) and (3) some of the error messages need work. Program compilation and execution are a bit slow, but not so slow as to be a problem in an introductory course.

The Grace-based class at Pomona was run as an experimental section of our introductory course in CS. All class materials, including lectures notes, assignments, and the draft of a textbook [2] can be found at the course web site at <http://www.cs.pomona.edu/~kim/CSC051GF14/>.

Because Grace is a much simpler language than Java, the course could move faster than the Java-based version. By 10 weeks into the course we were 1 week ahead of the Java-based sections. Students did essentially the same labs in both the Java and Grace-based sections. While no statistical comparisons were made, student performance on the midterm and labs in the Grace section were at least as strong, and perhaps stronger, than in the Java sections,



© Kim B. Bruce and Andrew P. Black;  
licensed under Creative Commons License CC-BY

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

even though we moved at an accelerated pace. Because the students in the experimental section needed to transition to a second course that assumed they knew Java, the last four weeks of the course were spent learning Java, as well as covering other new topics. We will be tracking those students this term to see how they fare in the follow-up course.

We noticed several places where the simplicity of Grace simplicity allowed us to spend less time on annoying details, and thus focus more of our attention to the fundamentals of learning to program. These included Grace having a single numeric type, no primitive types (everything is an object), appropriate visibility defaults, object literals rather than special constructor methods for classes, no type-based overloading, extensible lists rather than fixed-length arrays, and no null value. The tutorial we wrote for Grace programmers transitioning to Java makes many of these differences clear: it is available at <http://www.cs.pomona.edu/classes/cs051G/handouts/GraceToJava.pdf>.

### 3 Next steps

Another course using Grace, though with more advanced students, is currently in progress at Portland State. This is in preparation for a course with beginning students that will be taught during the summer. The Pomona introductory course will be taught again next fall, during which we will make a careful comparison of the Grace-based and Java-based versions.

### 4 Summary

Learning to program is hard; no programming language is going to change that. However, we believe that a modern programming language designed for novices can greatly reduce the accidental complexity of learning to program. We hypothesize that students with a thorough grasp of the basics of object-oriented programming will not have much trouble moving to an industrial OO language; early experience at Pomona seems to support this hypothesis.

We are interested in hearing what the PL community thinks about attempts to design this kind of language. Is there support for attempts to move introductory courses to new languages (like Grace and Pyret<sup>1</sup>) that are designed for teaching novices?

**Acknowledgements** We thank the other members of the Grace team, particularly James Noble, Michael Homer, and Tim Jones, without whose support we would never have been able to run these trial classes.

---

#### References

- 1 Andrew P. Black, Kim B. Bruce, Michael Homer, James Noble, Amy Ruskin, and Richard Yannow. Seeking Grace: a new object-oriented language for novices. In *SIGCSE*, 2013.
- 2 Kim B. Bruce, Andrea Danyluk, and Tom Murtagh. Programming with Grace, 2014. <http://www.cs.pomona.edu/~kim/ProgrammingWithGrace/bookmain.pdf>.
- 3 Michael Homer, Timothy Jones, James Noble, Kim B. Bruce, and Andrew P. Black. Graceful dialects. In Richard Jones, editor, *ECOOP 2014 — Object-Oriented Programming—28th European Conference, Uppsala, Sweden.*, volume 8586 of *LNCS*, pages 131–156. Springer, 2014.

---

<sup>1</sup> <http://www.pyret.org>