Teaching with Grace

Kim B. Bruce[1]

1 Department of Computer Science, Pomona College
Claremont, CA 91711, USA
kim@cs.pomona.edu

1 Grace

Grace [1] is a new object-based object-oriented language designed for novices. While we are still tweaking some aspects of the language design, and the current implementations are still rather rudimentary, we used Grace in an introductory course at Pomona College this fall for the first time. Designed simply as a trial run to find out what aspects of the language and implementation needed work, very positive student feedback leads us to consider this first trial a great success. We anticipate working on the implementations and development environment over the next year and then performing a careful experiment to compare the Grace version of our introductory course with the present Java-based version of the course.

In a one-page abstract, the best we can do to describe Grace is to list buzz-words: object-oriented, object-based, first-class closures (anonymous functions), dialects for programming levels, gradually typed, with a straight-forward syntax and semantics. For details, we recommend you visit www.gracelang.org, which includes links to papers on the Grace language design.

2 Teaching Gracefully

The implementation of Grace used for the class was web-based, with program editing, compilation, and execution all performed in a web browser window. This had the great advantage that no software had to be installed on students' computers, and the overhead of learning to use these tools was quite low. The main weaknesses of this implementation are (1) the gradual (and even static) type system isn't yet fully supported, (2) there is no support for threads (timers are used instead to support animations), and (3) the quality of error messages needs to be improved. Program compilation and execution is a bit slow, but not enough to be a problem in an introductory course.

The Grace-based class was run as an experimental section of our introductory course in CS. All class materials, including lectures notes, assignments, and the draft of a text book [2] can be found at the course web site at http://www.cs.pomona.edu/~kim/CSC051GF14/. The fact that Grace is a much simpler language than Java meant that the course could move at a faster clip than the Java-based version. By 10 weeks into the course we were 1 week ahead of the Java-based sections of the course. Students did essentially the same labs in both the Java and Grace-based sections of the course. While no careful normed comparisons were made, student performances in the Grace section on the midterm and labs were at least as strong, and perhaps stronger than the Java sections, even through we moved at an accelerated pace. A careful comparison between the Grace-based and Java-based versions of the course will be made during the fall semester this year.

Examples of places where Grace's simplicity allowed less focus on annoying syntactic issues included a single numeric type, no primitive types (everything is an object) appropriate visibility defaults, object expressions, no separate constructors for classes, no static

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

overloading, support for lists rather than primitive arrays, and no null value. (The tutorial for Grace programmers to learn Java makes clear many of these differences: `http://www.cs.pomona.edu/classes/cs051G/handouts/GraceToJava.pdf`)

3 Summary

Programming 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 inessential complexity of learning to program, while making it easy to move to a more industrial-strength language. What does the PL community think about attempts to design this kind of language? Are members willing to support attempts to move introductory courses to new languages (like Grace, Pyret `http://www.pyret.org`, etc.) that are designed for teaching novices?

Acknowledgements Thanks to the Grace language team: Andrew Black, James Noble, Michael Homer, and Tim Jones, without whose support we would never have been able to run this trial class.

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 *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 129–134, New York, NY, USA, 2013. ACM.

2 Kim B. Bruce, Andrea Danyluk, and Tom Murtagh. *Programming with Grace*. http://www.cs.pomona.edu/ kim/ProgrammingWithGrace/bookmain.pdf, 2014.