nated and students were required to complete only a single report.) Finally, the professor began creating extensive course notes to compensate for the lack of a suitable text.

The student evaluations following these changes are also found in Figure 3. The overall rating of the course improved significantly, without sacrificing the course's organization or feedback from the students' perspective. Furthermore, it is clear that, while students still found the workload heavier than in other courses, the situation had improved significantly from the previous year. The student comments likewise supported these changes.

- *"Considering this was the first time almost any of us have used Java (or even object-oriented type of code), the class was quite good. The tutorials were a great idea... I also liked the fact that we learned through examples."*
- *"Excellent course. Java is a LOT more interesting than other languages we have learned."*
- *"I liked that you used lots of chemical engineering-relevant examples... it made it a lot easier to see just how OOP works and what it can do."*

LOOKING BACK AND LOOKING FORWARD

Object-oriented methodologies offer several significant advantages when designing and building process simulators. After two years of teaching these approaches to fourth-year chemical engineering students at the University of Ottawa, it appears that most of these students are able both to grasp the basic concepts of object-oriented programming and to apply these to design problems within a single semester course. Further, it is clear that Java provides an effective tool with which students can explore these concepts. Success, however, also depends to a great extent on the opportunities for practical application of object-oriented programming through structured tutorials, assignments, and projects.

A course of this nature continues to face challenges. First and foremost, chemical engineering students must be exposed to object-oriented design within the context of processes and systems familiar to them—there is little benefit in illustrating object-oriented programming in Java with a bubble-sort routine. Unfortunately, while there is a plethora of Java-related texts available on the market, none of these is well suited to a course in computer-aided design. This gap must then be filled by well-conceived examples and detailed notes from the instructor.

While some engineering schools have introduced object-oriented languages such as Java and C++ during the sophomore year of their undergraduate program, the engineering program at the University of Ottawa continues to teach a procedural language (in this case, C) to most of its undergraduates. Therefore, when undertaking the computer-aided design course described here, our students must first learn the basic syntax of Java. More importantly, they must also rethink the programming process, often "unlearning" approaches adopted during their study of procedural languages. Both of these activities limit the time available within this single semester course to present more complex examples of object-oriented process simulation.

ChE letter to the editor

Dear Editor:

Query: A colleague told me that the following equation

$$f = 1.375 \cdot 10^{-3}\left[1 + \left(20,000\frac{\varepsilon}{D} + \frac{10^6}{\text{Re}}\right)^{1/3}\right]$$

reproduces the turbulent part of the friction factor chart very well. I have tested it; it does. It is much easier to use than any other such function that I know of, because it is explicit in each of the three variables.

Alas, my colleague does not know where it came from. I would like very much to know the benefactor of our profession who invented this function, and I would like to give credit when I cite it.

If any of our readers can tell me who the author is and cite a proper reference for this function, I will be most grateful.

Noel de Nevers
University of Utah
<Noel.deNevers@m.cc.utah.edu>

A more effective approach, in the author's opinion, would be to offer a programming course in Java to all engineering students during their first year of study, leaving more time to focus on practical applications of the language in later stages of the undergraduate curriculum.

ACKNOWLEDGMENT

The author wishes to acknowledge the chemical engineering students at the University of Ottawa who have proven themselves, time and again, to be equal to the challenge.

REFERENCES

1. Booch, G., *Object-Oriented Analysis and Design with Applications*, Addison-Wesley (1994)
2. Horstmann, C.S., and C. Cornell, *Core Java 1.2, Volume I-Fundamentals*, Prentice Hall (1999)
3. Kafura, D., *Object-Oriented Software Design and Construction with C++*, Prentice Hall (1998)
4. Holman, J.P., *Heat Transfer*, McGraw Hill (1990).
5. Taylor, D.G., "Computer-Mediated Collaborative Learning in ChE at the University of Ottawa," *Chem. Eng. Ed.*, **33**, 250 (1999) ❐