

# Fourth “Killer Examples” for Design Patterns and Objects First Workshop

Carl G. Alphonse  
University at Buffalo  
alphonse@cse.buffalo.edu

Stephen Wong  
Rice University  
swong@cs.rice.edu

Michael E. Caspersen  
University of Aarhus  
mec@daimi.au.dk

Adrienne Decker  
University at Buffalo  
adrienne@cse.buffalo.edu

## Introduction

We are proposing to organize the “*Killer Examples*” for *Design Patterns and Objects First* workshop for a fourth time. The first two organizers have been involved with the workshop since its inception at OOPSLA 2002 in Seattle; Michael Caspersen was a participant at both the 2003 workshop at OOPSLA 2003 in Anaheim and the 2004 workshop at OOPSLA 2004 in Vancouver.

The last three year’s workshops were successful on several fronts. The reviews of the workshops by the participants were all very positive. Participants indicated they would like to see future workshops on the same topic and would recommend the workshop to a colleague. Last year we experimented with a half-day workshop, but feedback from participants have convinced us to return to a full-day format.

We have disseminated results from the workshops in several ways. We have presented posters at both OOPSLA and SIGCSE:

- Carl Alphonse, D.X. Nguyen, Phil Ventura and Stephen Wong. (2004). “OOPSLA 2004 “Killer Examples” Workshop: Exploring “Killer Examples” for teaching design patterns and objects-first”. A poster presentation at the *19<sup>th</sup> Annual Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, Vancouver, Canada.
- Carl Alphonse, Stephen Wong, D.X. Nguyen, Phil Ventura and Michael R. Wick. (2003). “The “Killer Examples” for Design Patterns and Objects First workshops held at OOPSLA 2002 and 2003”. A poster presentation at the *18<sup>th</sup> Annual Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, Anaheim, CA.
- Carl Alphonse, D.X. Nguyen, Phil Ventura and Stephen Wong. (2003). “Teaching Object-orientation and Design Patterns in CS1/CS2 with Killer Examples”. A poster presentation at *The 34<sup>th</sup> Technical Symposium on Computer Science Education (SIGCSE)*, Reno, NV.

We also presented a hands-on workshop at SIGCSE 2004 which presented results from the first two OOPSLA “Killer Examples” workshops. The fact that the workshop filled to capacity (30) and had a waiting list of 16 demonstrates that there is great interest in techniques and methods for teaching OO at the introductory course levels. The OOPSLA workshops have a different focus than the SIGCSE workshop had; the OOPSLA workshop serves as a forum for educators with successful techniques and approaches to share them with each

other. The SIGCSE workshop sought to disseminate the results of the OOPSLA workshop more widely, to educators who are seeking effective means to teach this material.

- Carl Alphonse, Dung X. Nguyen, Phil Ventura, Michael R. Wick and Stephen Wong. (2004). “Teaching Design Patterns in CS1/CS2”. A workshop at the 35<sup>th</sup> *Technical Symposium on Computer Science Education (SIGCSE)*.

Due to the success of the previous years’ workshops we are proposing to run it for a fourth year, with modifications to take into account our experiences at the previous OOPSLA workshops and the SIGCSE workshop.

## Themes and Goals

**killer app** The application that actually makes a sustaining market for a promising but under-utilized technology.

### The Jargon File

The theme of this workshop is “killer examples” for design patterns and objects-first. We take these to be examples which provide clear and compelling motivation for a particular design pattern or object oriented concept.

Today’s computer science students are tomorrow’s software developers. They will design and implement solutions to complex problems. Object orientation is an excellent approach to managing the complexity of large, real-world software systems. Design patterns are an essential part of an object oriented approach to managing complexity. Our experience teaching CS1/CS2 has convinced us that it is possible and desirable to teach object orientation from a design patterns perspective.

Faculty and software developers who have learned to program using a non object-oriented approach often view object orientation with skepticism. Some find it difficult to see the benefits that object orientation offers. Others recognize that object orientation has great promise but feel that both object orientation and design patterns are too complex and too abstract for beginners to learn.

The objects-first approach is recognized in CC2001 [0] as a legitimate entry point to the computer science curriculum. There are compelling arguments [1,2,3] for the inclusion of design patterns at an early stage of the computer science curriculum, even in an objects-first CS1 course. Our collective experience teaching CS1 and CS2 has convinced us that it is possible and desirable to teach object orientation from a design patterns perspective.

A question to be addressed is how to present design patterns and object oriented concepts in a grounded manner so that their purpose and applicability are plainly grasped by a beginning student. There is certainly no dearth of materials expounding the wonders of design patterns (e.g. [5]). These materials are, for the most part, directed at those with considerable experience with object orientation to start with. Many object-oriented programming texts lack original compelling and grounded examples to serve as motivation for fundamental object oriented concepts. An excellent text which suffers this fate is [4]: its presentation of design patterns is tied to the GOF examples and applications within the JFC/Swing classes. More variety and inspiration would surely help students understand the significance and practical usefulness of design patterns.

In [2], the authors conclude by noting that since “Design patterns and object orientation do not always scale down, care in choosing examples will help ensure that educators and students appreciate the power of design patterns.”

It is the aim of this workshop is to address this issue. We believe there are “naturally occurring” and compelling examples to motivate individual design patterns. The goals of the workshop are

1. to elicit from the participants “the application [the example] that actually makes a sustaining market [compelling argument] for a promising but under-utilized technology [design patterns],” and
2. to bring together educators and developers to share their respective design pattern and object-oriented expertise, and
3. to provide a forum for discussion of techniques for presenting these examples and the design patterns they showcase to students.

## References

- [0] Computing Curricula 2001, Computer Science Volume. <http://www.acm.org/sigcse/cc2001/>
- [1] Dung Nguyen and Stephen Wong. Design Patterns: Pedagogical Foundations For Object-Orientation. A workshop presented at *University of Wisconsin System Computer Science Education Workshop*, University of Wisconsin, October 13, 2000. <http://exciton.cs.oberlin.edu/uwisconsin/>
- [2] Owen Astrachan. OO Overkill: When Simple is Better than Not. Appeared in the *Proceedings of the Thirty Second SIGCSE Technical Symposium on Computer Science Education*, pages 302–306. 2001.
- [3] Owen Astrachan, Geoffrey Bernt, Landon Cox and Garrett Mitchener. Design Patterns: An Essential Component of CS Curricula. Appeared in the *Proceedings of the Twenty Ninth SIGCSE Technical Symposium on Computer Science Education*, pages 153–160. 1998.
- [4] Cay Horstmann. *Object-Oriented Design & Patterns*. John Wiley & Sons, Inc., 2004.
- [5] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

## Organizing Committee

- Carl Alphonse, Research and Teaching Assistant Professor, University at Buffalo.  
e-mail: [alphonce@cse.buffalo.edu](mailto:alphonce@cse.buffalo.edu)
- Stephen Wong, Lecturer, Rice University.  
e-mail: [swong@cs.rice.edu](mailto:swong@cs.rice.edu)
- Michael E. Caspersen, Director, IT-Vest, University of Aarhus.  
e-mail: [mec@daimi.au.dk](mailto:mec@daimi.au.dk)
- Adrienne Decker, Lecturer, University at Buffalo.  
e-mail: [adrienne@cse.buffalo.edu](mailto:adrienne@cse.buffalo.edu)

## Primary Contact and Chair

Carl Alphonse  
201 Bell Hall  
Department of Computer Science & Engineering  
University at Buffalo, SUNY  
Buffalo, NY 14260-2000

e-mail: [alphonc@cse.buffalo.edu](mailto:alphonc@cse.buffalo.edu)  
phone: (716) 645-3180 x115  
fax: (716) 645-3464

## Organizer Backgrounds and Roles

**Carl Alphonc** is Research Assistant Professor and Teaching Assistant Professor at the University at Buffalo, State University of New York. He earned a Ph.D. from the University of British Columbia in 2000. He has taught an object-oriented introduction to computer science for almost 7 years.

### *Background*

He has organized and been chair of the first, second and third “Killer Examples” workshops at OOPSLA 2002, 2003 and 2004. He organized and was chair of the “Teaching Design Patterns in CS1/CS2” workshop at SIGCSE 2004. He organized and was chair of a special session entitled “Teaching using off-the shelf on-line materials” at SIGCSE 2001. He was co-organizer of a session at the 2002 International Conference on Engineering Education.

Selected publications:

C. Alphonc and P. Ventura. “Using graphics to support the teaching of fundamental object-orientation principles in CS1,” 2003. Proceedings companion of the 18th Annual Conference on Object-Oriented Programming, Systems, Languages and Applications, ACM Press.

C. Alphonc and P. Ventura. “QuickUML: A Tool to Support Iterative Design and Code Development,” 2003. Proceedings companion of the 18th Annual Conference on Object-Oriented Programming, Systems, Languages and Applications, ACM Press.

C. Alphonc and P. Ventura, “Object-orientation in CS1-CS2 by design,” 2002. SIGCSE Bulletin 34:3, 70-74.

P. Ventura and C. Alphonc, “Teaching OOD and OOP through Java and UML in CS1 and CS2”. Position paper presented at the *Fifth Workshop on Pedagogies and Tools for Assimilating Object Oriented Concepts*, October 2001.

<http://www.cs.umu.se/~jubo/Meetings/OOPSLA01/Contributions/PVentura.html>

### *Workshop role*

His roles in the workshop include chair and primary contact. As chair he is primarily responsible for the organization of the workshop, putting out the call for submissions and participation, coordinating the submission reviewing process, maintenance of the listserv for pre- and post-workshop activities, maintenance of the workshop web site, scheduling the workshop activities, and introducing and running the workshop. He will also contribute to the creation of a poster reporting on the results of the workshop.

**Stephen Wong** is a Lecturer at Rice University. He earned a Ph.D. from MIT in 1989. He developed and implemented curriculum for an introductory computer science course which utilized an innovative objects-first-with-design-patterns methodology at Oberlin College in 1998, and has taught object-orientation ever since.

### *Background*

He organized a workshop with Dung Nguyen, entitled “Design Patterns: Pedagogic Foundations of Object Orientation.” He has been a member of the organizing committee of the “Killer Examples” workshops at OOPSLA 2002, 2003 and 2004, and also at the “Teaching Design Patterns in CS1/CS2” workshop at SIGCSE 2004.

He has published numerous papers on applications of design patterns as well as the teaching of design patterns. Selected publications:

D. Nguyen and S. Wong, "Design Patterns for Parsing," Proceedings of the 36th SIGCSE Technical Symposium, 2005.

<http://www.exciton.cs.rice.edu/research/SIGCSE05>

D. Nguyen, M. Ricken and S. Wong, "Design Patterns for Marine Biology Simulation", Proceedings of the 35th SIGCSE Technical Symposium, March 2004, 467-471.

<http://www.exciton.cs.rice.edu/research/SIGCSE04>

D. Nguyen and S. Wong, Design Patterns for Self-Balancing Trees. OOPSLA 2002 Educator Symposium, November 2002, 60-70.

<http://exciton.cs.rice.edu/research/OOPSLA02/>

D. Nguyen and S. Wong, "Design Patterns for Games," Proceedings of the 33rd SIGCSE Technical Symposium, March 2002, 126-130.

<http://www.exciton.cs.rice.edu/research/SIGCSE02>

D. Nguyen and S. Wong, "OOP in Introductory CS: Better Students Through Abstraction". Position paper presented at the *Fifth Workshop on Pedagogies and Tools for Assimilating Object Oriented Concepts*, October 2001.

<http://www.cs.umu.se/~jubo/Meetings/OOPSLA01/Contributions/swong.pdf>

D. Nguyen and S. Wong, "Design Patterns for Sorting," SIGCSE Bulletin 33:1, March 2001, 263-267.

<http://www.exciton.cs.rice.edu/research/SIGCSE01>

D. Nguyen and S. Wong, "Design Patterns for Lazy Evaluation," SIGCSE Bulletin 32:1, March 2000, 21-25.

<http://www.exciton.cs.rice.edu/research/SIGCSE00>

D. Nguyen and S. Wong, "Design Patterns for Decoupling Data Structures and Algorithms," SIGCSE Bulletin 31:1, March 1999, 87-91.

<http://www.exciton.cs.rice.edu/research/SIGCSE99>

### ***Workshop role***

His roles in the workshop include reviewer and moderator. As reviewer he is responsible for reviewing submissions for possible acceptance. As moderator he is responsible for facilitating discussion on the listserv during pre-workshop activities as well as during the workshop itself. He is also primarily responsible for creating a poster reporting on the results of the workshop.

**Michael E. Caspersen** is an Academic Researcher and Director of IT University West at University of Aarhus, Denmark. He has a B.Sc. in computer science and mathematics from University of Aarhus (1984) and a M.Sc. in computer science from University of Aarhus (1987). He has taught (introductory) programming for 20 years and (introductory) object-oriented programming for 15 years.

### ***Background***

He organized and was chair of ITiCSE 2002 conference in Aarhus, and he has been on the program committee of ITiCSE 2000, ITiCSE 2001, and the 2nd, 3rd, and 4th Annual Finnish/Baltic Sea Conference on Computer Science Education (2002-04). Together with Michael Kölling he organized and held a full-day workshop entitled "Teaching Introductory Object-Oriented Programming – Dangers, Traps, and a Road Map" at SIGCSE 2001, and a similar tutorial at ITiCSE 2001. Together with Jens Bennedsen he organized and held a workshop entitled "Model-Driven Programming Education" at SIGCSE 2005. He was a participant of two previous "Killer Examples" workshops, at OOPSLA 2003 and 2004.

He has published more than a dozen papers on computer science education, including papers on how to teach frameworks and design patterns early in the curriculum. He is also author of a two-volume textbook on programming (in Danish, 1993). Selected publications:

Caspersen, M.E.; Christensen, H.B.: "The Need for Killer Examples for Object-Oriented Frameworks". In: Proceedings of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2003, Anaheim, California, USA, 26-30 October 2003.

Caspersen, M.E.; Christensen, H.B.: "Frameworks in CS1 – a Different Way of Introducing Event-driven Programming". In: Proceedings of the seventh Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2002, Aarhus, Denmark, 24-26 June 2002.

Caspersen, M.E., “A C++ Framework for Active Objects in Embedded Real-Time Systems – bridging the gap between modeling and implementation”, In: Technology of Object-Oriented Languages and Systems, TOOLS 32, Melbourne, Australia, pp. 52-65. IEEE Computer Society, 1999. ISBN 0-7695-0462-0.

### ***Workshop role***

His roles in the workshop include reviewer and moderator. As reviewer he is responsible for reviewing submissions for possible acceptance. As moderator he is responsible for facilitating discussion on the listserv during pre-workshop activities as well as during the workshop itself. He will also contribute to the creation of a poster reporting on the results of the workshop.

**Adrienne Decker** is a Lecturer at the University at Buffalo, State University of New York. She holds a Master’s Degree in Computer Science and Engineering from the University at Buffalo, and is currently working on the completion of her Ph.D., which is focused on assessment issues in introductory computer science courses.

### ***Background***

She has taught an object-oriented introduction to computer science at UB since 2001 and has been involved with the continued improvement of the curriculum in those courses as well as the creation of assignments that blend the introductory material with key object-oriented concepts and especially with the use of design patterns.

Selected publications:

Ventura, P., Egert, C., and Decker, A. (2004) “Ancestor Worship in CS1: Reexamining the Introduction of Arrays”, 2004 OOPSLA Educator’s Symposium, Vancouver, BC, pp. 68-72

Decker, A. (2003) “A Tale of Two Paradigms”, The Journal of Computer Sciences in Colleges, Vol 19 Issue 2, pp. 238-246.

### ***Workshop role***

Her roles in the workshop include reviewer and moderator. As reviewer she is responsible for reviewing submissions for possible acceptance. As moderator she is responsible for facilitating discussion on the listserv during pre-workshop activities as well as during the workshop itself. She will also contribute to the creation of a poster reporting on the results of the workshop.

## **Previous related workshops**

The most directly related prior workshops are the previous “Killer Examples” workshops. The proposed workshop follows the same pattern as the previous three, addressing the feedback of participants.

We have selected two workshops/conferences as representative of a larger variety of events related to design patterns. We contrast the focus of these with that of the workshop we are currently proposing.

- *Workshop on Pedagogies and Tools for Assimilating Object Oriented Concepts*

This recurring workshop has as its goal for to “share experiences about alternative teaching approaches and tools to improve the teaching and learning of the basic concepts of object technology rather than teaching a specific programming language.”

<http://oopsla.acm.org/oopsla2001/fp/workshops/05.html>

This workshop has in the past alternated between OOPSLA in North America and a conference in Europe, such as ECOOP. The last North American hosting of the workshop occurred in 2001, at the Tampa OOPSLA.

- *ChiliPLoP*

ChiliPLoP is a conference “designed to meet the special needs of two groups in the software community, pattern newcomers and pattern experts.”

<http://www.agcs.com/supportv2/techpapers/patterns/chiliplop/>

Our current workshop will bring together educators and developers to address how best to ground abstract design patterns in a manner which makes them accessible to those unfamiliar with object orientation and/or design patterns.

Our workshop addresses a more specific concern (examples for design patterns) than the *Workshop on Pedagogies and Tools for Assimilating Object Oriented Concepts*. It also addressed a broader audience as it seeks to involve both academics and developers.

The *ChiliPLoP* conference focuses on writing patterns, by providing proving feedback and support for developers who author patterns. Our workshop is searching for examples which motivate the use of existing design patterns.

## Expected number of participants

In the first two workshops we restricted participation to only those who submitted examples for presentation. Based on our experience with the SIGCSE workshop enrollment, as well as inquiries we have received regarding “observer status” at the previous OOPSLA workshops, we understand there is interest in participating without presenting. We therefore made last year’s workshop more accessible in two ways:

- We encouraged submissions to participate as *presenters* and as *discussants*. A presenter submitted a “Killer Example”, while a discussant submitted a statement of interest. Prior to the workshop everyone (presenters, discussants and organizers) communicated via a listserv mailing list, discussing drafts of the examples to be presented at the workshop. At the workshop presenters presented their examples and participated in the discussion of other examples, while discussants participated in the discussions only.
- We also allowed walk-ins at the workshop, as long as there was capacity in the room (we reached the room capacity of about 20 - we had expected about 12 participants).

We propose to structure this year’s workshop like last year’s, with presenters and discussants, as well as walk-ins. The main distinction between a discussant and a walk-in is that discussants are guaranteed place in the workshop, and can participate in the pre-workshop activities.

We expect 3 to 5 example presenters, and 6 to 10 other participants. Including the four organizers the workshop would have between 13 and 19 total participants. The ideal number would be 4 example presenters and 8 discussants, for a total of 16 total participants.

## Workshop preparation (pre-workshop activities)

Prior to the workshop we expect participants to submit their contribution for posting on a web page. As with previous workshops, a listserv e-mail forum will be set up to allow for pre-workshop discussion of the posted proposals. We will invite participants to discuss the problems, their solutions, as well as the contributor’s justification for the example. We expect the discussion to cover the effectiveness of the submitted examples and what makes examples effective (or ineffective). Constructive feedback to improve submitted examples will also be welcome, so that the workshop discussions can focus on more substantial issues.

## Planned workshop activities

We are proposing a full-day workshop. The workshop program will consist of a short introductory presentation by the organizers, followed by presentations by the presenters of their submissions. Here is a tentative schedule for events, assuming we have 4 accepted presentations:

**8:00 Introduction** 10 minutes

**8:10 Presentations of two examples** 90 minutes (30 minutes for presentation, 15 for questions/discussion, for each example)

**9:40 Coffee break** 20 minutes

**10:00 Discussion period** 90 minutes

**11:30 Lunch** 90 minutes

**1:00 Presentations of two examples** 90 minutes (30 minutes for presentation, 15 for questions/discussion, for each example)

**2:30 Coffee break** 20 minutes

**2:50 Discussion period** 90 minutes

**4:20 Workshop wrap-up (summary of findings, reactions)** 10

This schedule assumes that the day starts at 8:00 AM and finishes at 4:30PM. If the day is shorter we can shorten lunch and breaks. If the day is longer we can distribute that time to presentation and/or discussions periods. Experience from the previous workshops has shown that 60 minutes for lunch is a little bit tight, especially if restaurants close to the workshop venue are crowded with other workshop attendees.

## Post-workshop activities

After the workshop participants will be expected to submit final versions of their examples for inclusion on a web site for general dissemination. The web sites from previous years' workshop are linked from :

<http://www.cse.buffalo.edu/faculty/alphonse/KillerExamples/>

As noted, we have in the past presented posters of the workshop at the conference poster session, and are prepared to do so again. The workshop poster is halfway prepared ahead of time, with general information about the workshop and its presentations, but with open space for workshop outcomes, to be filled in after the workshop.

The organizers will also endeavor to advertise the workshop activities by submitting to relevant related conferences, such as SIGCSE 2006 and OOPSLA 2006. We maintain, at the above-noted website, materials from previous years' workshops which serve as a resource for educators seeking materials. We will encourage participants to post updates to their materials after the workshop is over and as they are using the examples in the classroom.



## Special requirements

The workshop organizers and most participants will bring laptops. We will therefore require power for them. *A data projector has been available for each of the previous workshops, and has proven very beneficial.* We note the “NOTE” in the call for workshop proposals that “There will be no data projectors available for Workshops. Due to contractual reasons it’s not possible to bring your own data projector.” We would be most pleased if arrangements for data projectors could be made (e.g. is it possible to secure a data projector for the workshop if the organizers institutions share the cost?) If data projectors are in no way possible, we will suggest alternate means of presenting the examples (e.g. posters, which can perhaps be incorporated into the workshop poster).

Other items which we have made use of in the past include an overhead projector, easels with paper and markers, and whiteboards with dry-erase markers. These will be essential if a data projector is not available.

## Abstract

According to the Jargon File, a **killer app** is an “application that actually makes a sustaining market for a promising but under-utilized technology.” A “killer example” provides clear and compelling motivation for a design pattern or design principles. An example is “killer” if it compels someone to spontaneously “buy in” to that design pattern or technique.

“Killer examples” are important pedagogically because they get students on-board and excited about design in general and design patterns in particular. They can also motivate design pattern use to those not familiar object-orientation, dispelling the myth that design patterns are an esoteric topic.

The workshop’s goal is to gather educators and developers who have “killer examples” to share, or who wish to participate in the discussion of the examples/techniques for presenting design patterns to beginning students. Pre-workshop activities encourage interaction and refinement of examples prior to the workshop.