

Contact Us!

Teachers: “How can I learn to teach the BJC course?”

Administrators: “How can I get BJC into my schools?”

Businesses: “How can I support computer science education in my area?”

Send email to bjc@bspace.berkeley.edu

We will bring BJC teacher preparation to your location, if you have about 20 teachers willing to commit a full eight weeks to this demanding course. Our funding covers salary and expenses for an experienced BJC teacher to spend two weeks on-site, one before and one after the participants take the BJC course itself online. (All course materials, including lecture videos and Moodle lab exercises, are available free at <http://bjc.berkeley.edu>)

Our funding does not include stipends for the participating teachers. We suggest that administrators seek local business funding to support the teachers who will be giving up their summers to this program.

We are funded for three summers of this program. If we are successful at getting BJC taught in the participating school districts, we plan to apply for additional funding to expand the program. NSF wants to prepare 10,000 *new* high school computer science teachers by 2015, so we are especially eager to work with teachers in districts that do not already have a high school CS curriculum in place.

Our model is to partner with faculty and students at local colleges who can provide a meeting room, teaching assistantships, and continuing support during the school year following the summer preparation. If you have such a relationship already available, let us know. If not, we will help seek out interested faculty at a college near you. We will also help you join or establish a local chapter of the Computer Science Teachers Association (<http://csta.acm.org>).

If you are an isolated teacher who can't find a big enough group to support this program, email us anyway; we may be able to connect you with other interested teachers in your area.

Dan Garcia and Brian Harvey, University of California, Berkeley
Tiffany Barnes, University of North Carolina, Charlotte



<http://bjc.berkeley.edu>
bjc@bspace.berkeley.edu



The Beauty and Joy of Computing BJC

A new computer science course for high school or undergraduates

Highlights:

- Programming in 
- Social Implications of Computers in 
- Meets *AP CS: Principles* standard
- Attract nontraditional CS students
- *Free* Professional Development!
- All course materials *free* online.



Topics include abstraction, graphics, programming paradigms, games, algorithms, concurrency, recursion, privacy, copyright, artificial intelligence, game theory, higher order functions, limitations of computing, and the future of computing.

AP CS Principles, CS10K, and the NSF CE21 Initiative

BJC was chosen as one of the initial pilots for a **new “AP CS: Principles” exam** to be introduced in 2015. The purpose of this course is to attract nontraditional computing students (especially women and minorities, but also English majors) to the breadth and depth of ideas in modern computer science.

The National Science Foundation wants to prepare 10,000 new high school computer science teachers to teach the new AP course by 2015. Under their CE21 (Computing Education for the 21st Century) initiative, we have been funded to provide **free intensive eight-week summer workshops** for high school teachers, including two weeks of face-to-face training, one before and one after six weeks of our online course.

Testimonials from students

“This course is AMAZING!”

“This course is so practical and useful!”

“This course was one of the best, most interesting, and most challenging classes I’ve ever taken, ever.”

“This course was my favorite course this semester. Because of it, I am considering a major in CS.”

“This course was enlightening. I never thought I’d take a computer science class, ever. I’m glad there’s an introductory course like this that demystified how computers work on a basic level. It also made me realize how computing has and will continue to change the world.”

Please join us at these SIGCSE events:


Thu 5:10pm: BJC BOF

Fri noon: Free SNAP! Lunch

Fri 7pm: BJC workshop (FREE for HS teachers)

Sat 3pm: SNAP! workshop

Inside...

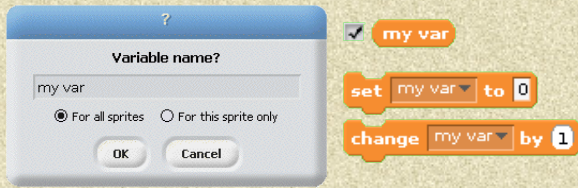
1. The  programming language
2. The *Blown to Bits* textbook
3. The *Blown to Bits* textbook
4. **Contact us!** Get free professional development for BJC teachers.

Funded under NSF CE21 grants 1138596, 1143566

SNAP! was previously known as BYOB (Build Your Own Blocks).

Some Big Ideas in Scratch

• Variables and Scope



• Parallelism



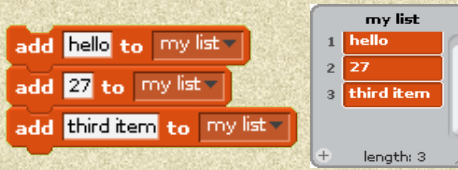
• Event Handling



• Iteration



• Lists

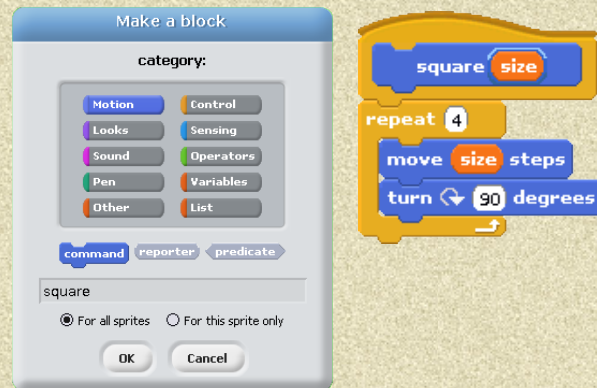


• Broadcast Messages

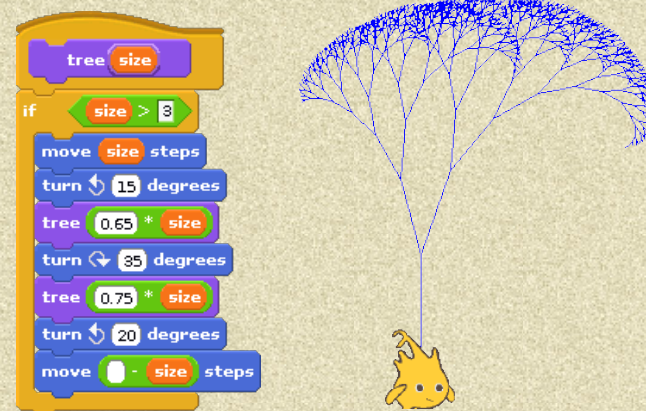


Big Ideas Added in λSnap! (<http://snap.berkeley.edu>)

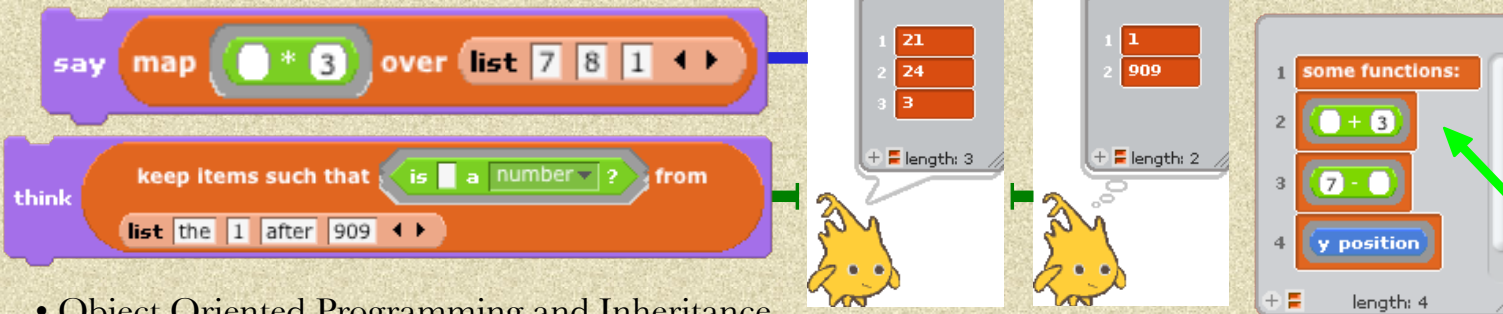
• Procedures (Build Your Own Blocks)



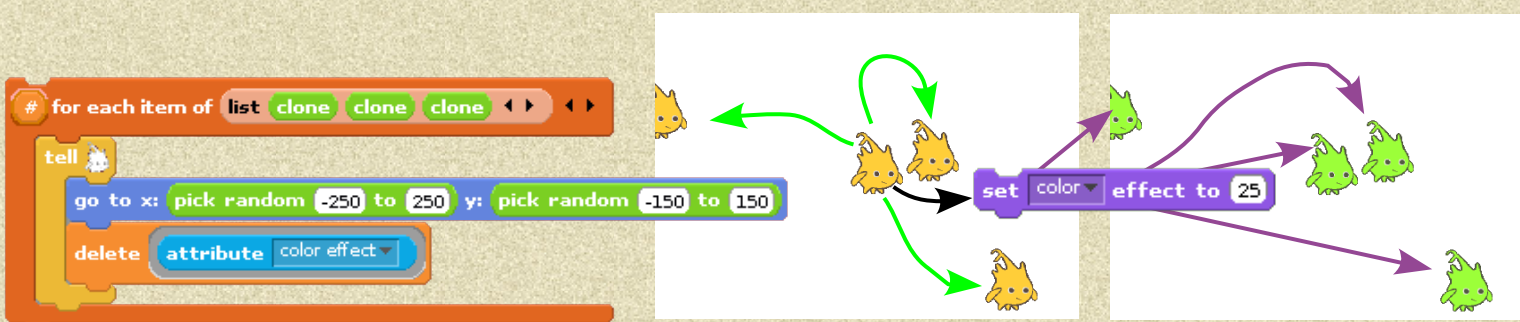
• Recursion



• Functions and Lists as First Class Data



• Object Oriented Programming and Inheritance



Blown To Bits

Your Life, Liberty, and Happiness after the Digital Explosion

The textbook used in BJC, available *free* online, *Blown to Bits* anchors the non-programming, social aspects of the course.

Implications of Digital Revolution

"Who owns all the data about us?"

"Which news sources can we trust?"

Ten paradoxical digital data truths that are overturning centuries-old assumptions.

Accessible Technology Explanations

Binary data, compression, search, cryptography, networking and others explained simply. For example, the book describes Google's PageRank algorithm in enough detail for order of growth analysis, with no assumptions about the reader's programming experience.

Praise for Clarity and Timeliness

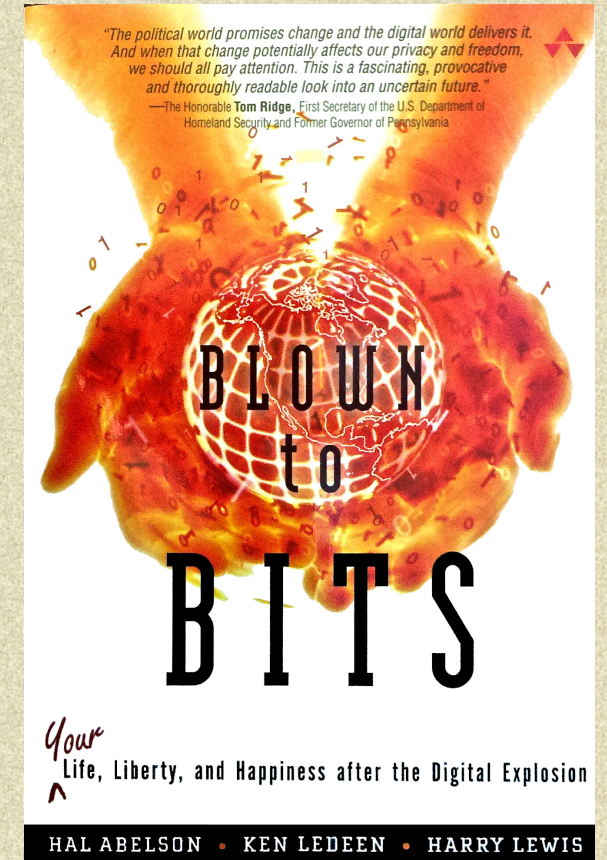
"There is no simpler or clearer statement of the radical change that digital technologies will bring, nor any book that better prepares one for thinking about the next steps. Rich in examples and beautifully accessible in its writing, it should be required reading everywhere."

— Lawrence Lessig, Professor of Law, Stanford University

Authors: Hal Abelson, Ken Ledeen, Harry Lewis


Cost: Free online PDFs (hardcover \$31 list, \$21 typical online)

URL: <http://bitsbook.com>



"Why not use Java/Javascript/Python/Processing/Scheme?"

Scratch (<http://scratch.mit.edu>) has had enormous success in getting child programmers up to speed instantly, with no anxiety, partly because syntax rules are visually apparent: A block can be used in a slot if the two are the same shape. Also, no keyboard skills are required. For these and other reasons, several recent CS 0 curricula start with Scratch and then switch to a text-based language. But SNAP! has *both* the intuitive Scratch interface *and* the full power of any other programming language. And the visual presentation helps make advanced abstractions such as function-as-data concrete.

Compare:  vs. `(lambda (x) (+ x 3))` or `function (x) {return x+3}.`

SNAP! runs in any browser—no software download/install necessary.