

AP CS Principles: Beauty and Joy of Computing Syllabus (ID#: 1648113v1)



Course Overview:

BJC covers the entire CS Principles Framework and addresses the seven Big Ideas in the framework with a primary emphasis on programming (Big Idea 5) and, closely linked with programming, on abstraction (Big Idea 2). As much as possible, BJC uses programming as the vehicle to tell other parts of the story; for example by presenting data (Big Idea 3) not through commercial database software but by writing programming projects that manipulate data sets as lists.

The secondary emphasis of BJC is on the social implications of computers (Big Idea 7, Global Impact). Social topics are included in every unit, not just one. Students are encouraged to think critically about each application of technology. The Big Ideas of creativity (Big Idea 1) and algorithms (Big Idea 4) are addressed throughout the units, and there is particular attention to the Internet (Big Idea 6) in Unit 4.

Snap!, the programming language used in BJC, was developed specifically for this curriculum. Its visual, drag-and-drop design is based on that of Scratch, so that it is accessible to a wide audience and not intimidating, but the language, itself, is extended with the abstraction mechanisms needed for serious computer science: first class procedures for control abstraction and first class lists for data abstraction. These capabilities are embodied in carefully chosen visual metaphors so that ideas traditionally considered difficult can be understood and enjoyed by beginners.

Reference Text:

Blown to Bits. Abelson, H., Ledeen, K., & Lewis, H. 2008, Addison-Wesley.

Students do not need their own copy of this book; text is available free online.

Primary Course Resources:

- Snap! (snap.berkeley.edu/run)
- Beauty and Joy of Computing curriculum (bjc.edc.org)

Teaching Strategies:

The Beauty and Joy of Computing (BJC) curriculum is designed to provide students with the opportunity to develop computational thinking practices and an understanding of the social implications of computing and to engage in programming. There is a strong focus on learning through labs and projects where students have opportunities to create programs and other computational artifacts. Students are engaged in small group and whole class discussions, particularly in relation to social implications topics. The curriculum content is designed to foster collaboration and creativity. Throughout the course, students engage in pair programming, and there are ample opportunities for students to collaborate and be creative in learning the course content.

Assessment:

Students are primarily evaluated on the basis of their work, which can take the form of worksheets, writing assignments, and programs. Self-check quizzes are built into the Beauty and Joy of Computing lab pages, and end-of-unit assessments help to prepare students for the AP Create and Explore through-course assessments.

AP Through-Course Assessment: *Explore - Impact of Computing Innovations* performance task

- After completing Unit 4, students complete through-course assessment *Explore - Impact of Computing Innovations* (8 hours in class).

AP Through-Course Assessment: *Create - Applications from Ideas* performance task.

- After completing Unit 6 and before the AP exam, students complete through-course assessment *Create - Applications from Ideas* (12 hours in class).

Units of Study:

Unit 1: Introduction to Snap! Programming

Students dive right into Snap! programming by designing games and visual projects using loops, randomness, and control blocks. They get to know the Snap! interface, coordinate system, sounds, mathematics, variables, and how to build their own blocks. Students are encouraged to *experiment* with Snap!, playing with inputs, trying out blocks before they have been formally introduced, and modifying and extending the assigned tasks.

Students discuss social implications of computing, including video games, privacy, and an introduction to the general question of whether or to what extent technology is good or bad.

Unit 2: Conditionals, Abstraction, and Debugging

Starting with the challenging task of teaching the computer to generate the plurals of nouns (e.g., butterfly → butterflies, moth → moths, bush → bushes), students begin thinking about the *structure of programs*. They learn to test for special cases, and how to use, sequence, and optimize the conditionals that control a program based on those cases. They also begin to think about what makes correctly working code into “good” code—is it the brevity, the clarity, or some combination? They learn to think about debugging by deliberately looking for ways to make a program fail and then finding ways to avert failures.

Students consider the innovations around us that collect data about us, and the ways in which that information may be available online. They discuss why we generally want to protect privacy and for what purposes we might choose to give up some privacy. They learn how they can protect their own online privacy. Students also examine communications technology and ways that computing impacts community (including cyberbullying).

Unit 3: Lists

This unit introduces lists, a powerful data type for storing multiple items of any type, including numbers, strings, other lists, or even blocks (function/procedure as data). Just

as functions can take numbers and strings as inputs, they can also take lists as input or produce lists as output. A list is an ordered sequence of items.

Students are introduced to several powerful higher-order list-processing functions that take functions, along with other data, as inputs. One virtue of the higher order list functions is that they generate new lists to report rather than mutate existing lists. This is in contrast with the imperative, looping, mutation-based programming that is more common, but more error-prone, in dealing with sequential data. Near the end of the course, students build these higher order functions for themselves.

The social implications topics in Unit 3 are *search* and *encryption (part 1)*. In both cases, the readings from *Blown to Bits* describe the underlying technology and also raise questions about risks. The BJC lab pages focus more on the latter. In the case of searching, the main issues are about user profiling (for serving advertising and for tailoring search results) and about bias in results (profile-based or otherwise). For encryption, we do emphasize how revolutionary public key cryptography has been, but the main emphasis is on how the various social stakeholders view the question of strong encryption of user data.

Unit 4: The Internet and Global Impact

Unit 4 addresses the structure of the Internet, the various protocols on which it runs, and the implications of this technology to society. Students learn to recognize HTML and learn that it is another computer language, and they learn to “scrape” an HTML page for data. They build on their work in Unit 3 to analyze lists containing data using Snap! procedures and strategies they have already learned.

Students first consider what rules perhaps ought to exist regarding behavior or content on the Internet. They use the *Blown to Bits* reading and what they've learned about the Internet throughout this Unit to discuss the challenges of regulating the Internet, look at ways that countries (including the United States) approach regulation and censorship of online content, examine statistics about Internet usage around the world.

Students continue to familiarize themselves with the AP Explore Task requirements, specifically by examining an innovation that addresses a social issue to which they feel some personal connection. They first identify social issues they care about and learn some ways that technology has been used to address the issue and then use the AP Explore Task prompts to guide them towards thinking and writing about potential benefits and possible unintended negative consequences of the innovation.

Unit 5: Algorithms and Data

This unit focuses on several types of analysis: analysis of problems to generate algorithms for their solution; analysis of the algorithms (especially of the time it takes to execute them) in order to optimize them; analysis of phenomena by generating models and simulations that give insight and help one generate and test hypotheses; and analysis of data, especially including visualization.

Students have been generating algorithms to solve problems from the start of this course, but have not yet focused on analyzing them for efficiency. For small enough computational problems, such analysis isn't needed. But modeling complex phenomena and handling large data sets requires understanding that there are sometimes alternative

algorithms that reduce the impact of the size of a model on the time it takes to execute. In-depth coverage of this broad domain (computational complexity, data analysis, modeling and simulation) is beyond the scope of an introductory course, but Unit 5's projects in each of these areas will give students a good first-approximation understanding of these issues.

Students learn the meaning of copyrights and patents, discuss their impact and relevance in an increasingly computationally mediated world, and learn about the simultaneous invention of the telephone.

Note: At the end of Unit 5, all of the CSP curriculum framework has been addressed. Units 6 and 7 contain additional material that's important to BJC including recursion and functional programming. Part of Unit 6 and all of Unit 7 will come after the AP exam in May.

Examples of Opportunities to Meet Learning Objectives Connected to Computational Thinking Practices

P1: Connecting Computing

- In Unit 4, Lab 6, *Internet Security*, students learn about common security attacks (bug exploits, viruses, phishing, and DDoS attacks) and learn about security measures that they can take to mitigate their own risk. LO 6.3.1 [P1]
- In Unit 4, Lab 8, *Innovating for Social Change*, students research and analyze the contextual suitability and effects of a computing innovation. LO 1.2.5 [P4], LO 7.1.1 [P4], LO 7.3.1 [P4], LO 7.4.1 [P1], LO 7.5.1 [P1], LO 7.5.2 [P5]

P2: Creating Computational Artifacts

- In Unit 1, Lab 1, *Building an App*, students collaborate to develop a program in Snap! to execute algorithms that control sprite appearance and movement in order to implement the desired behavior in a simple game. LO 4.1.1 [P2], LO 4.1.2 [P5], LO 5.1.3 [P6]
- In Unit 2, Lab 3, *Developing More Complex Programs*, students use an iterative and exploratory development process and multiple levels of abstraction as they create scripts to draw a single tile, a row, and a grid for a game board. LO 1.1.1 [P2], LO 2.2.2 [P3]

P3: Abstracting

- In Unit 1, Lab 2, *Sprite Drawing and Communication*, students analyze how several programs implement algorithms, e.g. discussing how a short algorithm with a loop generates the image of a square; comparing and contrasting various square-drawing algorithms; and predicting what an algorithm does. LO 5.2.1 [P3]
- In Unit 5, Lab 6, *Traffic Simulation*, students develop a simulation to model the phenomenon of highway traffic and use their model to formulate, refine, and test hypotheses about the context. LO 2.3.1 [P3], LO 2.3.2 [P3]

P4: Analyzing Problems and Artifacts

- In Unit 2, Lab 1, *Conditional Blocks*, students analyze a script to determine what is likely to go wrong before running it and use this developing understanding to

create correct programs to calculate powers of numbers and do a piece of linguistic research, counting the number of vowels in a chunk of text. LO 5.4.1 [P4], LO 5.1.2 [P2]

- In Unit 2, Lab 4, *Privacy; Community and Online Interactions*, reflecting on their own data and technology, students extend their insights to consider how large data sets of many people's data affect large-scale communication, interaction, collaboration, and problem-solving. LO 3.2.2 [P3], LO 7.1.1 [P4], LO 7.1.2 [P4]

P5: Communicating

- In Unit 4, Lab 4, *Network Protocols*, students learn about IP address hierarchy, IPv4 vs. IPv6, packets and packet switching, reliable data transmission, open standards, the Internet abstraction hierarchy and how these systems interrelate. Students write a simple program to report their IP address by scraping a webpage, explore a simulation of unreliable data transmission, and discuss and write about the abstractions of the Internet including protocols and address hierarchy. LO 6.1.1 [P3], LO 6.2.1 [P5], LO 6.2.2 [P4]
- In Unit 5, Lab 2, *Graphing Data and Functions*, students create a general graphing program that plots data points on a screen whose dimensions (scale) they have determined themselves. Students interpret and communicate the results of their data-processing by using appropriate visualizations, notations, and precise language. LO 3.1.3 [P5]

P6: Collaborating

- In Unit 3, Lab 1, *Introduction to Lists*, students build a basic shopping list app and learn that lists can store data and that programs can access and manipulate list contents. Students collaborate to create a computational artifact using lists and list-processing procedures to organize information. LO 1.2.2 [P2], LO 5.1.3 [P6]
- In Unit 4, Lab 5, *Weather App*, students collaborate to create a new computational artifact using the `http` procedure together with a weather website and then selecting and communicating specific information to the user. LO 1.2.2 [P2], LO 1.2.3 [P2], LO 3.1.2 [P6], LO 3.1.3 [P5]

Examples of Opportunities to Meet Learning Objectives Connected to Big Ideas

Big Idea 1: Creativity

- In Unit 2, Lab 2, *Script Variables*, students collaborate to create a program—a computational artifact—for creative expression in art, language, or mathematics by using Snap! and the graphics and information-processing techniques they have learned. LO 1.2.1 [P2], LO 1.2.4 [P6], LO 1.3.1 [P2]
- In Unit 3, Lab 5, *Search*, students consider questions like: How important is privacy to me? What sources of information might be overlooked by the engineers who design these search engines at large companies? What phrases have special meaning to me that Yahoo or Google don't seem to know about? LO 1.2.5 [P4]

Big Idea 2: Abstraction

- In Unit 3, Lab 2, *Nesting Lists*, students work with data abstraction at multiple levels, creating, using and modifying abstract data types. For example, they create

- a `get_new_contact` procedure that calls both a `get_name` and a `get_phone` procedure; they also work with a list of items whose elements are pairs of coordinates stored as an abstract data type `point` that students create. **LO 2.2.1** [P2], **LO 2.2.2** [P3], **LO 2.2.3** [P3]
- In Unit 4, Lab 3, *Number Representation*, students learn about bits and bit width and representations of digital data, and they translate among three common number bases. **LO 2.1.1** [P3], **LO 2.1.2** [P5], **LO 5.5.1** [P1]

Big Idea 3: Data and Information

- In Unit 2, Lab 4, *Privacy; Community and Online Interactions*, reflecting on their own data and technology, students extend their insights to consider how large data sets of many people's data affect large-scale communication, interaction, collaboration, and problem-solving. **LO 3.2.2** [P3], **LO 7.1.1** [P4], **LO 7.1.2** [P4]
- In Unit 4, Lab 2, *GPS Data*, students collaborate as they process GPS data, find patterns to gain new knowledge, and discover and explain patterns and trends. **LO 3.1.1** [P4], **LO 3.1.2** [P6], **LO 3.1.3** [P5], **LO 3.2.1** [P1], **LO 4.1.1** [P2], **LO 5.1.1** [P2]
- In Unit 5, Lab 2, *Graphing Data and Functions*, students interpret and communicate the results of their data-processing by using appropriate visualizations, notations, and precise language. **LO 3.1.3** [P5]

Big Idea 4: Algorithms

- In Unit 5, Lab 3, *Timing Experiments*, students write a timer program that accepts a function (and inputs) and times the computation in milliseconds. Students recognize *constant* time processes, *linear* time, *quadratic* time, and *logarithmic* time and evaluate algorithms' efficiency, correctness, and clarity. **LO 4.2.1** [P1], **LO 4.2.4** [P4]
- In Unit 5, Lab 4, *Unsolvable and Undecidable Problems*, students learn the difference between solvable and unsolvable problems and about the existence of undecidable problems through work with the halting problem. **LO 4.2.2** [P1], **LO 4.2.3** [P1]

Big Idea 5: Programming

- Students program throughout, beginning in the first lesson. In Unit 3, Lab 3, *Three Key List Operations*, students use `map` (which applies an input function to each item of a list) to transpose music, perform geometric transformations on shapes, perform algebraic functions on list items, etc. They use `keep` (which filters a list) to select desired list items. And they use `combine` (which combines list items using an input binary operation) to sum, average, or find the maximum value in a list of numbers. These higher order functions are themselves abstractions. **LO 5.3.1** [P3], **LO 5.5.1** [P1]
- In Unit 4, Lab 2, *GPS Data*, students collaborate as they process develop an algorithm to process GPS data, find patterns to gain new knowledge, and discover and explain patterns and trends. **LO 3.1.1** [P4], **LO 3.1.2** [P6], **LO 3.1.3** [P5], **LO 3.2.1** [P1], **LO 4.1.1** [P2], **LO 5.1.1** [P2]

Big Idea 6: The Internet

- In Unit 3, Lab 6, *Encryption*, students explore the basic programming concepts involved in cybersecurity and conduct online research requiring the evaluation of sources for appropriateness and credibility. **LO 6.3.1 [P1]**, **LO 7.5.2 [P5]**
- In Unit 4, Lab 4, *Network Protocols*, students learn about IP address hierarchy, IPv4 vs. IPv6, packets and packet switching, reliable data transmission, open standards, the Internet abstraction hierarchy and how these systems interrelate. Students write a simple program to report their IP address by scraping a webpage, explore a simulation of unreliable data transmission, and discuss and write about the abstractions of the Internet including protocols and address hierarchy. **LO 6.1.1 [P3]**, **LO 6.2.1 [P5]**, **LO 6.2.2 [P4]**

Big Idea 7: Global Impact

- In Unit 1, Lab 6, *Explosion of Bits; Games and Violence*, students analyze the effects of computing as they debate the impact of violent video games on youth violence; consider the implications of data practices (e.g. privacy, free-speech, etc.) involved in managing and manipulating people's data; and connect computing to social and cultural contexts. **LO 3.3.1 [P4]**, **LO 7.3.1 [P4]**, **LO 7.4.1 [P1]**
- In Unit 5, Lab 7, *Copyrights and Patents*, students consider the good or bad effects of streaming software such as BitTorrent, the benefits of Creative Commons and other free licensing schemes, and how technology such as region coding on DVDs affects the global digital divide. **LO 6.3.1 [P1]**, **LO 7.2.1 [P1]**, **LO 7.4.1 [P1]**