

To Use or Not to Use? Unsolicited Comments on “The Labs”

*By Maria Litvin, Phillips Academy, Andover, Mass
and
Gary Litvin, Skylight Publishing*

Note:

January 29, 2014

Since this document was first created, the AP CS Exam Development Committee has made the following announcement:

More Information About the AP CS A Lab Requirement for 2014-15

Dear AP Computer Science A teachers:

In October 2013, we announced that, starting in the 2014-15 school year, the GridWorld case study will be replaced by a required lab component consisting of three new AP Computer Science A labs.

To ensure that teachers have greater flexibility of instruction, **the required lab component has been changed to a minimum of 20 hours of hands-on lab experiences**, which can be fulfilled with either the new AP Computer Science A labs or other, comparable labs teachers wish to use. GridWorld will still be removed from the 2014-15 AP Computer Science A curricular requirements.

- The three new AP Computer Science A labs **will not be required** and will be provided as exemplars to support the implementation of at least 20 hours of hands-on lab work in the classroom.
- Teachers will have the flexibility to modify and implement these new labs according to the needs of their students and to use other resources to meet the 20-hour lab requirement.
- The 2015 AP Computer Science A Exam will not have specific questions that focus on the new AP Computer Science A labs and will continue to assess the concepts and skills outlined in the Course Description. In addition, there will no longer be any GridWorld case study questions.

The question before us is how (or whether) the three labs put forward by the Exam Development Committee can be used effectively in an AP CS A course. In previous years, the case studies, especially GridWorld, were used to help teachers focus on and teach certain topics and as assessment tools for AP exams. They made sure students gained experience working with a fairly large software project and were prepared to answer questions about that project.

The purpose of the labs is less clear at the moment. One of the intentions is perhaps to make sure that an AP CS course includes an adequate lab component. This doesn't seem to be a big problem, since most textbooks and instructional materials already include many labs and programming exercises, and most teachers already have their students work on lab assignments.

The main problem with the labs, as we see it, is their sheer size: they are too extensive. The Student Guides, combined, have a total of 76 large pages, which is equivalent to about 120 textbook pages, 3-4 chapters. This is a lot of reading. The labs also involve hundreds of lines of code. According to the authors, completion of all of the activities in the labs may require 40 hours of class, which is about 10-12 weeks of class time, perhaps as much as one third of the total time available before the AP exam. If used for teaching, will the labs deliver proportional benefits? Are they the most effective way to teach the topics they intend to teach?

To answer these questions, let us consider each lab separately.

Magpie

This lab offers an opportunity to practice with nested `if-else` statements and reading Java API. But its main emphasis is on manipulating strings and using `String` methods. According to the author, “this can be done early in the course, as soon as students have learned `if` statements and start working with the `String` class.” (In our textbook, *Java Methods*, this happens in Chapter 10, after the students have learned the basics of syntax and style, classes and methods, algorithms, arithmetic, Boolean variables, relational and logical operators, `if` statements, loops, and more details on classes and got comfortable reading and modifying relatively long segments of code.) According to the author, completing the first four activities in the lab should take between 3.5 and 7 hours, that is one or two weeks. What do we gain for that time spent?

The current AP subset includes five `String` methods: `length`, `indexOf`, `substring(from, to)`, `substring(from)`, and `compareTo`. We believe these and some other methods can be introduced effectively in one class period, one lab, and a number of short, exercises — two or three class periods total. (In our textbook, we use a fairly short lab *Lipograms*, which involves operations similar to Magpie: extracting a word from a text string, replacing a substring in a string, concatenating strings, etc.). Magpie goes a little out of the AP subset, introducing `String` methods `toLowerCase` and `indexOf(fromPos)`. The author chose to stay within the subset and avoid using the `charAt` and `Character`'s `isLetter` methods when the need arose to determine the type a character in a string.

We estimate that in a busy course, the `String` “chapter” should take the total of one week of class. String handling is reinforced later in the course. Playing with chatbots will realistically take two weeks, and students will get an idea that NLP (Natural Language Processing) involves a lot of “attention to detail” (or drudgery, depending on one’s taste). This will leave out many important concepts and topics: immutability of strings, literal strings and escape characters, how `compareTo` and `compareToIgnoreCase` work, extracting numbers from strings and formatting numbers into strings, `Character` methods. We think at most one Magpie programming activity might be appropriate for an AP course. Some students, no doubt, will find chatbots a lot of fun; they can continue improving their chatbot on their own.

Picture Lab

This lab was developed by Barbara Ericson and is based on Mark Guzdial’s Media Computation approach to teaching CS. Mark and Barbara are authors of several books on Media Computation (all of which are listed on the lab’s bibliography page, including the Python version).

The focus of the lab is working with 2D arrays in the context of manipulating digital images. The lab also mentions in passing binary numbers —

“Computers use binary numbers, which use the digits 0 and 1 and powers of 2 to represent values using groups of bits. A bit is a binary digit, which can be either 0 or 1. A group of 8 bits is called a byte. The binary number 110 means 0 ones (2^0) plus 1 two (2^1) plus 1 four (2^2), for a total of 6.”

— and a paragraph on abstract classes and interfaces. The claim in the Teacher’s Guide that the lab teaches binary numbers and inheritance and interfaces appears to be an exaggeration.

The nine activities in the lab include understanding the RGB color representation, and handling 2D arrays: setting values of elements, copying regions within the same array and from one array to another, and so on.

~~The AP subset includes only rectangular 2D arrays, and students do not have to know that a 2D array is a 1D array of 1D arrays. The lab goes a little out of the subset, using nested “for each” loops to traverse a 2D array. It is not clear why this is necessary.*~~

According to the author, it may take anywhere between 5.5 and 15 hours to complete the nine main activities in the lab, which translates into 1.5 to 4 weeks of class time. A typical AP syllabus may allocate two weeks to one- and two-dimensional arrays, combined. It will need to address such topics as array declaration and initialization, accessing the size and elements, traversing, finding the sum, max and min, passing arrays to constructors and methods and returning arrays from methods, inserting and deleting values, declaring and initializing 2D arrays, traversals, and various algorithms and exercises. In our textbook, we devote to 2D arrays one section, one lab, and several exercises, 3-4 class periods total.

Manipulating digital images is great fun and is appropriate, time permitting, for an introductory CS course or computer club or summer camp, but its extensive use in an AP course raises serious doubts. We recommend that at most one activity is included (Chromakey, one of the two lab extensions, seems like most fun).

Elevens

The Elevens lab is different from the other two. While Magpie and Picture Lab focus, more or less, on a single topic (strings and 2D arrays, respectively), the Elevens lab covers a vast array of topics, including class design, [ArrayLists](#), array algorithms, Selection Sort, modeling, and software testing. This lab is in effect a case study in disguise; perhaps more extensive than GridWorld. According to the authors, the lab will take about 16 hours to complete, or 4 weeks of class time.

The problem with this is that the lab uses the context of a single card game (with extensions into similar games). We experimented with such approach in the very first edition of *Java Methods* (2001), where the theme of several labs and case studies throughout the book was *Ramblecs*, a game of our invention. We got mixed results: some teachers liked it, others thought “it’s all about one game.” Working with one card game throughout the course might become a distraction. Once started, students may want to finish it, but not have enough knowledge yet to do it successfully. (Also, some families and teachers may have objections to anything that has to do with card games.)

* Since this document was first written, the 2014-2015 Course Description came out, in which this aspect of 2D arrays has been added to the required knowledge and to the AP subset.

In our view, a project of this size and complexity is best left for a team project after the exam. In fact, one of us (Maria) used many times a similar team project based on the game of Set. It also involves classes for a card, a deck, and a board and implementing rules for finding a particular combination of cards on the board and replenishing the board (<http://www.skylit.com/projects/>).

Conclusion

In our opinion, one or two activities from Magpie and Picture Lab might fit into a typical AP CS A syllabus. Elevens is best left for a team project after the exam (unless the Development Committee decides to make it a new case study). Implementing the labs in their entirety in an AP CS A course would wreak havoc in the syllabus and may derail the course.

A Note on Women in CS

All of us are trying to do our best to attract more female students and underrepresented minorities to our field. However, this has to be done in meaningful ways. The Magpie lab includes a couple of references to “Women in NLP research,” while the Picture Lab includes a page on a prominent female computer scientist. We believe such items appropriately belong on classroom walls and in textbooks, but have no place in College Board’s exam materials; they will be shrugged off by most students, both male and female.