



Student Performance Q&A: 2003 AP[®] Computer Science A Free-Response Questions

The following comments on the 2003 free-response questions for AP[®] Computer Science A were written by the Chief Reader, Chris Nevison of Colgate University in Hamilton, New York. They give an overview of each free-response question and of how students performed on the question, including typical student errors. General comments regarding the skills and content that students frequently have the most problems with are included. Some suggestions for improving student performance in these areas are also provided. Teachers are encouraged to attend a College Board workshop, to learn strategies for improving student performance in specific areas.

Question 1

What was the intent of this question?

In Part (a) students needed to iterate over an array of type `College`, whose public interface was given, find a college whose name matched a given `apstring` parameter, and then change the tuition for that college. This involved searching an array and accessing and changing information for an array element using the given member functions. In Part (b) students were required to iterate over the same array of type `College`. They were to select and store in another array of type `College` those elements of the given array, a data member of the class `CollegeGroup`, which matched criteria given in the parameters (tuition in a range from low to high and region as given). Again, accessing information involved using the accessor member functions for the class `College`. This part also involved creating and storing items in a new array and resizing that array before returning it.

How well did students perform on this question?

The mean score was 4.76 out of a possible nine points. This question was more difficult than the A1 question often is, largely because, in addition to searching over an array, the information had to be accessed by using member functions. In Java this will always be the case; information for objects is always accessed and changed through methods (the term for member functions used in Java). The distribution of scores was bimodal, a mode at the high end and one at the low end. This indicated that the better students found this question easy, there were no tricky parts, and the algorithms were easy. However, the weaker students did poorly. It was hard for them to pick up partial credit if they did not understand the array manipulations and the object abstraction.

What were common student errors or omissions?

Weaker students made the typical array errors (e.g., incorrect bounds, off-by-one errors, and so on) and errors in the three-part Boolean expression. Other common errors were a poor understanding of accessing or changing information in an object via its member functions (methods in Java). There were syntax

errors in the combination of accessing an element of an array combined with the member function call. There was also a more fundamental error — students assumed a data representation for the `College` class (none was given) and tried to directly access the data as if they were data fields for name, tuition, and other fields.

Based on your experience of student responses at the AP Reading, what message would you like to send to teachers that might help them to improve the performance of their students on the exam?

Teach students to understand how to access information in an object using accessor methods and how to change it using mutator methods. Emphasize that, except in the definition of the class itself, the data of an object cannot be accessed or manipulated directly, only through the provided methods. This level of abstraction is at the heart of object-oriented programming. It is best taught by teaching the use of objects (from an appropriate library) from the first day of class and by getting into the definition of objects using class definitions early in the course. Students also need to understand basic array syntax and manipulation. In Java this will involve both Java arrays and the class `ArrayList`.

Question 2

What was the intent of this question?

This question involved writing member functions for a class `Company` that depended on using a given class `Employee`. Again, all access to information about the `Employee` class was through member functions that were given. Part (a) required students to write a Boolean member function that determined whether an employee satisfied two out of three criteria for retirement. Part (b) required students to remove all those elements of an array of type `Employee` that satisfied two of the three criteria (for which the Boolean member function would return `true`), leaving the remaining employees in the array in the same order (increasing by ID number).

How well did students perform on this question?

The mean score was 4.82 out of a possible nine points. Students did well with the algorithmic part of this question, but they had the same sorts of difficulties using the object abstraction for employees as they did on the first question. The overall results were slightly better than those for the A1 question because students generally did well on Part (a).

What were common student errors or omissions?

As with the A1 question, many students had difficulty with the use of objects and proper syntax and the use of calls to member functions. Students commonly removed items from the array in Part (b) in one of two ways: by 1) placing nonremoved items into an auxiliary array and copying back or by 2) sliding all remaining items in the array down to fill the vacated position every time an element was removed (both are N^2 solutions; few students found the efficient solution). These had the common errors for manipulating arrays, such as losing track of the correct position or off-by-one errors. Many students made Part (a), the Boolean function, more complicated than it needed to be or simply did not read the specification correctly and created a function that returned `true` only when all three conditions were `true` (instead of two out of three).

Based on your experience of student responses at the AP Reading, what message would you like to send to teachers that might help them to improve the performance of their students on the exam?

Students should be familiar with how to remove elements from an array, leaving the others in their original order. This is a common algorithm. As for the A1 question, students must know how to treat objects abstractly, using methods to access or modify information. This is fundamental to the object-oriented programming that will be the basis of programming in Java.

Question 3

What was the intent of this question?

This question tested students' understanding of working with two-dimensional arrays as represented by the class `apmatrix`. In Part (a) students were required to write a Boolean function based on checking whether given coordinates were in bounds on the array and the entry in the Boolean array was `true`. In Part (b) students needed to write a member function that examined the eight nearest neighbors of the given position in the two-dimensional grid in order to count the number of times the Boolean function from Part (a) returned `true`. In Part (c) students needed to scan over the entire grid encapsulated in the `TreasureMap` parameter and set the value in a parallel two-dimensional array to the number of neighbors for which the function in Part (a) returned `true`, or 9 if that function (`HasTreasure`) returned `true` for the location itself.

How well did students perform on this question?

The mean score was 5.28 out of a possible nine points. Students performed quite well on this question. This is the first time in many years that the two-dimensional array question has not been on both the A and the AB exams, and consequently its level was more appropriate for A-level students.

What were common student errors or omissions?

There were the common syntactic and logical errors about accessing elements in a two-dimensional array. Students who used loops to scan over the 3×3 block for Part (b) sometimes forgot to determine if the count needed to be adjusted for the center element (the easiest method was to count all nine positions and subtract one if the center position had a treasure). Students were sometimes confused about the use of the Boolean function `HasTreasure`. In Part (c) they sometimes missed the correct bounds for scanning over the two-dimensional array. Sometimes they did not realize that the function in Part (c) was a "free" function and not a member function of the class `TreasureMap`, and they therefore called the `TreasureMap` member functions incorrectly. (In Java there are no free functions, so the issue will always be which class includes the method being written and correctly accessing information for objects of other classes via their methods.)

Based on your experience of student responses at the AP Reading, what message would you like to send to teachers that might help them to improve the performance of their students on the exam?

Realize that two-dimensional arrays will no longer be a topic for the AP Computer Science A course. It will be a topic for the AB course. Nevertheless, manipulating elements within a two-dimensional grid based on row and column values will still be a topic. Indeed, the Marine Biology Simulation (MBS) case

study is based on such a model and is a likely source for such questions. However, the syntax for two-dimensional arrays is not part of the A course, so students will get information about a two-dimensional grid from a course that encapsulates the information and has methods that take either row and column parameters or position parameters that encapsulate row and column. The `Environment` class in the MBS case study is a good example. Students should understand the logic of working within a two-dimensional grid model and be aware that accessing information uses two indices, such as row and column, and that such indices themselves may be encapsulated in a class, such as the `Location` class in the MBS. Use the MBS case study throughout the course to introduce and give examples for topics such as these.

Question 4

What was the intent of this question?

This question was based on the Marine Biology Case Study (MBCS). Students needed to show their understanding of the case study and its interacting classes by writing member functions for the `Fish` class that depended on using member functions from the `Environment` class as well as its own member functions. Part (a) also required writing a very simple function for the `Environment` class that simply returned the value in the new `Algae` `apmatrix` at the corresponding position.

How well did students perform on this question?

The mean score was 2.25 out of a possible nine points. It was evident that many students had little or no knowledge of the MBCS and left the question blank or received very low scores. Those students who made a real attempt that indicated they had studied the MBCS did reasonably well.

What were common student errors or omissions?

Most students who understood the case study got Part (a) right. In Part (b) it was quite common for students to treat the `Neighborhood` object `nbrs` as if it was an array and use array syntax to access elements. In this part students were required to find the maximum in the `Neighborhood`. In doing so they also made typical errors, such as losing track of which was the index of the maximum and which was the value when updating the search or returning the position of the maximum. In Part (c) students made errors in the Boolean logic separating the three possible actions (eat, die, move). Few students correctly called the `Environment Update` member function when it was needed. It needed to be called if a fish ate or moved and therefore changed the value `myStepsSinceFed`, but not after it died (which would cause an error). Fortunately, in Java all object variables are references and the issues associated with having copies of fish in the case study (the Java MBS is similar to the MBCS) go away. This happens because the fish variables that appear in different contexts are always references to the one fish object at a given location. So there is no need for an update mechanism, except when the location of a fish is changed.

Based on your experience of student responses at the AP Reading, what message would you like to send to teachers that might help them to improve the performance of their students on the exam?

Teach the case study throughout the course. The new MBS case study for Java bears some similarity to the MBCS, but it also has many differences. It can be used to introduce or reinforce many topics in the A course and should be used throughout the course as these topics are introduced, not just at the end of the course. For example, the MBS gives excellent examples of interacting classes provided by a library (several of the MBS classes are, in effect, a library since students are not responsible for their implementations). The MBS gives good examples of how to modify existing classes to change the behavior of their objects. The MBS gives excellent examples of the use of inheritance to create subclasses with different behavior than the parent (super) class and uses of interfaces. These are topics in the A curriculum for Java.

Teach your students how to call methods for objects of one class from another class to create a program based on interacting objects of different classes. To facilitate this, *teach the use of objects and methods from the first day of class*. Also, teach about defining your own classes and programs built from interacting objects of different classes relatively early in the course (no later than halfway).

Addendum: Teachers should be aware of the changes in topics in the AP Computer Science A and AB curricula as AP Computer Science switches to Java and object-oriented programming. The best way to do this is to read the *AP Computer Science Course Description: May 2004, 2005*, including the sample questions, the course outline, and the commentary on the course outline.