Student Performance Q&A:

2014 AP® Computer Science A Free-Response Questions

The following comments on the 2014 free-response questions for AP® Computer Science A were written by the Chief Reader, Elizabeth Johnson of Xavier University of Cincinnati. 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?

This question asked the students to write two static methods for an unnamed class which had no instance variables. The question involved using methods of the `String` class (to get individual letters of the `String` and to build a new `String`) and methods of the `List` interface (to access elements of the `List` and to modify an existing `List`). In part (a), the students were required to implement a static method called `scrambleWord` to scramble a `String` parameter (`word`) by reversing pairs of adjacent letters of `word` whenever the pair consisted of "A" followed by a letter that was not "A". The students needed to access all letters in `word` from left to right, identifying the pairs of letters to be swapped. The identified pair could be reversed by rebuilding `word` or by constructing a new result `String`. All letters of `word` that were not involved in reversals were to be in their original positions in the result `String`, which was then returned by the method. In part (b), the students were required to implement a static method called `scrambleOrRemove`. The method had a `List` parameter (`wordList`) in which each element was a `String` of letters. The students needed to access each word in `wordList`, to correctly call the `scrambleWord` method of part (a) with the word as the argument, and to determine whether the word had been changed by scrambling. Upon exiting `scrambleOrRemove`, `wordList` was to contain only words changed by `scrambleWord` in their scrambled form. If the word had been scrambled by `scrambleWord`, the word was replaced in `wordList` by the scrambled version of the word. If the word was unchanged by `scrambleWord`, the word removed from `wordList`.

How well did students perform on this question?

This question appears to have been slightly more difficult than 2 and 4 but comparable to question 3. The mean score was 3.57 out of a possible 9 points with a standard deviation of 2.88.

What were common student errors or omissions?

In part (a), some students did not understand that strings are immutable and so tried to change in place. Students had difficulty identifying cases that needed to be covered in the algorithm. Some students tried to handle many more cases than necessary, creating opportunities for error. Students had difficulty with string methods such as `substring` and sometimes treated strings as arrays without converting to an array first.

In part (b), some students used the `==` operator rather than the `equals` method for comparison. Some students did not adjust the indices to take word removal into account, resulting in skipped entries. Some students also used an enhanced `for` (for-each) loop, which is problematic given the need for removal of words. Students also had difficulty on the syntax of calling a static method.

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?

Many students used types and methods outside of the AP® subset. The majority of these resulted in the loss of points for failure to name and use the appropriate types, methods, etc., correctly. Teachers should encourage their students to stay within the AP® subset for free-response answers. If teachers use chars in their classes, the difference between the values returned by concatenating chars and concatenating strings should be emphasized.

A review of early course materials as part of exam preparation might be appropriate. The need to use enhanced `for` loops only in situations where the underlying list will not be modified should be emphasized. Students should be reminded to use braces around indented blocks of code.

Students should be encouraged to take the time to read the questions carefully. Many students did not appear to fully understand the question before answering. Many students used Random to scramble words, looked for an `"A"`, but did not look for a non-`"A"` after it. Students should be reminded to carefully read and reread the question before writing their solution.

In part (b), students often tried to create a new list and then simply reassign the `wordList` parameter when finished. Students should be reminded that only the proper use of methods can modify the original object passed in as a parameter.

Question 2

What was the intent of this question?

This question involved reasoning in the context of the GridWorld case study. The question required the design and declaration of a class including proper keywords, inheritance, constructor, method overriding, and accessing and modifying inherited and nonmember instance variables and using constants. This problem tested students' knowledge of GridWorld classes/interfaces: `Rock`, `Actor`, `Location`, `Grid`, and Java classes/interfaces from the AP subset: `Object`, and `List<E>`. The question necessitated a class header, a no-argument constructor and overriding the `act` method. Students were required to create the `Director` class as a subclass of `Rock`. The question required a no-argument constructor, which set the color state of a director to be `Color.RED`. Students had to override the `act` method of the superclass so the same method signature and return type as the superclass (`Rock`) were required. The director's behavior required a check of the color state of the director in order to determine how to change colors. When the director starts to `act` in the green color state, the director will turn all of its neighbors to the right 90 degrees. The director turns (changes the direction of) all of its neighbors right by accessing the neighbor's current direction state in order to update and set the new direction state of the neighbor. If the

director's color state is green at the beginning of `act`, the director changes color to `Color.RED` and vice versa.

How well did students perform on this question?

This question appears to be of average difficulty. The mean score was 4.35 out of a possible 9 points with a standard deviation of 3.08.

What were common student errors or omissions?

Many students didn't consider that a `Director` could be any color other than red or green. As result, they often thought that `!RED` was sufficient for a check for `GREEN`. Many students didn't understand that the alternating color code should be done with if/else statements. They wrote if/if code instead that switched the color one way and then back to the original. Many students didn't understand that the color state of a `Director` was maintained in a companion `Actor` object. They maintained the color in a local variable instead of using the existing `Actor` color instance variable.

Many students did not know the correct form of calls to GridWorld methods. There was also confusion about the types of objects returned from calls. Some students used variables that were not declared. Students lacked an understanding about the state of an object over time.

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?

Teachers should spend more time on the proper way to write classes. Many students didn't seem to understand the basic anatomy of a class including `{}` placement, instance variables, and constructors.

Students also need to be taught about object state and how it is maintained in the object's instance variables, both in that object and the "companion" super-class object. For example, some students didn't understand how the director's color was maintained. Students also need more instruction on the implications of sequential *ifs* versus the *if-else* structure.

Question 3

What was the intent of this question?

This question involved the construction, initialization, and manipulation of a two-dimensional array. It also tests the student's ability to traverse a `List`, manage a counter, and return a value from a method. Students were asked to implement a constructor and a method of the `SeatingChart` class.

In part (a) students were asked to implement a constructor, which required the instantiation of the instance variable `seats`, a 2D array of `Student` objects, whose dimensions were determined by the parameters `rows` and `cols`. The elements of `studentList` were to be mapped to the 2D array `seats` in column-major order until all list elements had been assigned to the 2D array. Any remaining elements of the 2D array held their default `null` values.

In part (b) students were required to examine the instance 2D array `seats`, removing all `Student` elements whose absence count exceeded the parameter `allowedAbsences` by replacing the `Student` object with `null`. The method calculated and returned the number of `Student` objects that were removed.

How well did students perform on this question?

This question appears to be slightly more difficult than 2 and 4 but comparable to question 1. The mean score was 3.57 out of a possible 9 points with a standard deviation of 2.99.

What were common student errors or omissions?

Some students did not appear to read the question carefully and so omitted steps in the solution (create 2-D array, fill in column-major order, count, remove students). Many students failed to check for `null` before calling `getAbsenceCount()`. Some students had bounds errors in accessing the array. Some students were confused as to how and when to use enhanced `for` loops.

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 need more practice in looping over different data structures at the same time. Students also need more instruction on two-dimensional arrays. Students need more practice traversing arrays, making constructors, and checking boundary conditions.

Question 4

What was the intent of this question?

This question evaluated the ability of a student to read an interface and then define a class that implements that interface. As part of the implementation students would define a constructor with a specified order of parameters to instantiate a `Trio` object. Students were also required to include code to construct a `String` object from the supplied parameters of the component `MenuItem` objects within the `Trio`. This string was composed of the result of calling `getName()` on each `MenuItem` object in turn separated by "/" and ending with the `String` "trio". The resulting string was to be returned in the interface method `getName`. Students were also to compute the price of the `Trio` by identifying the lowest price of the three component items and excluding that amount from the total price of the `Trio` to be returned in the interface method `getPrice`.

How well did students perform on this question?

This question appears to be of average difficulty. The mean score was 4.14 out of a possible 9 points with a standard deviation of 3.44.

What were common student errors or omissions?

The most common student errors were omission of the instance variables, improper implementation of the interface methods and not being able to identify the lowest price correctly. Many students failed to provide implementation for the methods specified in the interface they are implementing. Some students are confused with implementing an interface and extending a class. Some students failed to notice that methods in an interface are automatically public. Some students made errors in creating a string from information in the object state.

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?

Teachers should continue to work with students on carefully reading the problem descriptions and answering within the scope of the question. Also, students should work on writing solutions which implement an interface and solve the stated problem. The solution that many students wrote was correct except for common errors in finding the minimum price and calculating the `Trio` price. Students should practice constructing correct conditional statements.