CollegeBoard
Advanced Placement Program

Student Performance Q&A:

2004 AP® Computer Science A Free-Response Questions

The following comments on the 2004 free-response questions for AP® Computer Science A were written by the Chief Reader, Chris Nevison of Colgate University. 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.

General Comments

The new exam introduced several new ideas. First, with Java as the programming language, an object-oriented approach to programming was expected. Most questions involved writing code within the context of class definitions for interacting objects. Students had to understand how to call methods for interacting objects from different classes. The use of methods is fundamental, so students were expected to call the appropriate method and not rewrite equivalent code.

A new type of question involving the design of classes was introduced on this exam (question 2). Students needed to understand inheritance and polymorphism to answer these questions correctly. For full credit students had to create a good design that followed the specification given in the problem.

Some general scoring principles are described on the "General Usage" sheet. Students were writing a draft solution under time constraints, so we did not penalize minor errors that did not reflect on the students' understanding. For example, confusion about the use of `length`, `length()`, and `size()` for accessing the length of a `String`, array, or `ArrayList` was not penalized. Other errors on the usage sheet indicate penalties taken if the error was not covered specifically in the grading rubric. Some minor errors were not penalized because a newer version of the language (Java 1.5) that allows different syntax is now available. These non-penalized errors included failure to downcast when removing objects from an `ArrayList` and failure to correctly convert between primitive types and their wrapper classes (e.g., `int` and `Integer`).

Question 1

What was the intent of this question?

This question tested students' understanding of simple algorithms on an `ArrayList`. Part (a) required a traversal through the list of `String` objects (words) to count the number with a given length. This Part tested a student's skill in looping through the structure, proper use of an `if` statement, and accumulation and return of the count. Part (b) asked students to search through the same `ArrayList` and remove those words with a given length. Since the structure was an `ArrayList` that re-indexes and resizes when an element is removed, this Part required recognition of the need to avoid skipping an element when the preceding element is removed. It also tested the loop and `if` logic.

How well did students perform on this question?

A surprising number of students (2,500 out of 13,800) did not answer or scored zero on this, the first (and easiest) question on the exam. This probably indicates a large number of students were not prepared for working with ArrayLists or were just not prepared for working with classes in Java. Those students who did respond did quite well on this question. They showed an understanding of the fundamental control logic of loops and conditionals, as well as an understanding of the use of `String` objects. The mean score was 5.8 out of 9.

What were common student errors or omissions?

Students often confused the `ArrayList` structure with an array, using array notation `myList[k]` instead of the correct `myList.get(k)`. Students also left out the needed downcast when getting an object from the `ArrayList`; however, this error was not penalized. Common logic errors included off-by-one bugs on the loop logic; reversed conditions in the conditional logic; and failure to declare, initialize, increment, or return a variable for counting. In Part (b) a common error was to use a `for` loop to iterate through the array without realizing that when an item is removed from an `ArrayList`, the structure is resized and elements after the element that was removed are re-indexed. Consequently, when two items in a row had the required length, the second would be skipped in the loop and never removed.

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?

Continue to teach simple algorithms that process information stored in array structures. Demonstrate the use of both arrays and the library structure `ArrayList`, including the differences in syntax for access and the resizing logic for an `ArrayList`. Students should understand the common process of correctly removing elements from an `ArrayList`.

Question 2

What was the intent of this question?

This question called for the design of classes within a given inheritance hierarchy. It was very similar to the sample problem about bank accounts in the course description. Students were expected to understand how a subclass inherits methods from a superclass so that they do not need to be redefined. The question also called for the definition of a method in a subclass that was specified as abstract in the superclass. Students needed to know how to access a superclass version of a method within the redefinition of the same method in the subclass using `super`. They also needed to know how to invoke the constructor from the superclass using `super`. Finally, students had to know how to use the polymorphism defined in the various subclasses in a client method that processes objects of the superclass type, `Pet`.

How well did students perform on this question?

Most students attempted this problem (there were only 750 no-responses and zeroes). Since it provided a template from which one could infer much of a solution for Part (a) and some for Part (b), most students got some points. However, the results were spread across the nine-point range, indicating a range of understanding of how to define a class, especially a subclass of a given abstract class. The mean score was 5.2 out of 9.

What were common student errors or omissions?

The most common misunderstanding was how to access information from the superclass within the definition of a method in a subclass. Many students did not understand 1) how to use `super` to invoke the super class constructor to set the instance variables declared in the superclass or 2) how to use `super` to invoke a method from the super class, as required in the `LoudDog speak` method which needed to invoke `super.speak()`. When students did not access the superclass methods properly, they would attempt to access the private instance variables inherited from the superclass—a major error. Some students would redefine the instance variables (for example, `myName`) in the subclass, setting its value in the constructor and using it in an override of the method `getName`. Even if this gives the appearance of correct function, it makes the inheritance hierarchy fragile in that changes in the superclass will not be propagated. Students who made this error did not understand how the subclass inherits methods from the superclass, meaning they should not be defined in the subclass, and how the superclass constructor must be invoked to set values for private instance variables declared in the superclass. Weaker students missed points on Part (c) by incorrectly accessing the elements of the `ArrayList` (using array notation) or for not understanding the polymorphism and trying to work around it.

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?

Inheritance and polymorphism are new and important topics for object-oriented programming. They are implemented in a relatively straightforward way in Java, but many students seem not to have learned the proper use of these tools. Teachers must show how to define subclasses within an inheritance hierarchy and what should and should not be defined in subclasses. Students should understand how to extend an abstract class. (On future exams they may be required to implement an interface.) Teachers must also demonstrate the use of polymorphism—for example, for the speak method in this problem.

Question 3

What was the intent of this question?

This question was based on the Marine Biology Simulation case study. It tested students' abilities to add new functionality within the context of a large program that they have studied. This new functionality was defined by a new class, `PondStocker`, which could be used in a redefined Simulation class (not part of the problem). Students had to understand the relationships among the existing classes in the simulation and how to use these classes in defining the methods of the new class. One method involved implementing a mathematical calculation, a second the use of random numbers, and a third the application of these two to changing the content of the environment in the simulation model.

How well did students perform on this question?

In the past, a significant number of students did not attempt the case study question, indicating that they hadn't studied it adequately in their school course. This year there were fewer no-response and zero scores on this question than on either question 1 or 4, indicating that only those students who were poorly prepared for the exam did not attempt the question. It seems that teachers are teaching the case study much better than they did in the past.

Very few students got full credit for this problem, because there were a couple of half-point deductions that were quite common. However, the distribution of scores across the remaining range of scores was quite even, indicating that students otherwise performed well. The mean score was 4.0 out of 9.

What were common student errors or omissions?

Part (a) of this question called for the calculation of the minimum number of fish that makes the density greater than a given value. A calculation must be done as a double, truncated and one added. Students found many ways to do this and often lost the half point for missing the correct rounding adjustment or other points in this computation.

Part (b) of the question called for the return of a random location from the environment. Students often did not use the `RandNumGenerator.getInstance` method to create the needed instance of `Random` (the purpose of this was explained in the case study). This showed that some understanding of the case study was needed. Students often attempted to use a method from the `Random` class without creating an instance of `Random`. In addition, some students misinterpreted the intent of this method, returning a random *empty* location, which is not in the specification (the check for empty belongs in the method defined in Part (c)). Students need to carefully read the specification for each method.

Part (c) called for the correct number of fish to be added to empty locations. Students often forgot to check whether a location was empty, sometimes got the syntax of accessing the environment incorrect, and often made the error of adding a fish twice. Because the `Fish` constructors add the fish to the environment, it is an error to construct a fish and then add that fish to the environment.

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 thoroughly (most teachers are doing this). Include examples of exercises like those given in the case study involving modification of the `Fish` class or the definition of subclasses of `Fish`, as well as examples that involve the definition of a new class that adds functionality, as called for in this problem. Be sure that students understand how to make use of the methods of classes given in the case study in this context.

Question 4

What was the intent of this question?

This question tested students' abilities to implement a relatively complicated algorithm. It used an array as the data structure, so students needed to know how to work with an array as opposed to the `ArrayList` used in question 1. Part (a) tested students' understanding of Boolean expressions and methods.

How well did students perform on this question?

This was the most difficult question on the exam, with more students with no-response and zero scores

(3,250) and many scores of 1 (1,100). Among other students the results were good, with more scores in the 6 to 9 range than in the lower range. The mean score was 4.4 out of 9.

What were common student errors or omissions?

In Part (a) students would sometimes be off by one when checking the high end of the array or would only check position and not direction.

In Part (b) students often mixed the logic of the conditionals. For example, they would not check for the tile being empty after removing an item, or they would change direction and location on the same move. There were several variations on how this logic can be incorrect.

In Part (c) students would sometimes reproduce code to check that the hall was clear. Although this can work out correctly, students would lose points for reproducing code when a simple call to the `hallIsClear` method is the correct way to check.

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?

Show students how to work with Booleans and `boolean` methods. Teach students to implement complex algorithms from a given specification. Perhaps most important, emphasize that students should not reproduce code for a method with the same functionality that is already given—even though such code may work, it is poor programming practice.