Student Performance Q&A:

2009 AP® Computer Science A Free-Response Questions

The following comments on the 2009 free-response questions for AP® Computer Science A were written by the Chief Reader, Jody Paul of Metropolitan State College in Denver, Colorado. 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 focused on the array data structure, its construction and traversal, the application of basic algorithms, and method invocation for a specified object. Students were provided with the framework of a helper class, `NumberCube`, that represented a conventional six-sided die (a cube with the numbers 1 to 6 on its sides). They were asked to implement two static methods of unspecified classes. In part (a) students were required to implement the `getCubeTosses` method that returns an array of values obtained by invoking the `toss` method of a `NumberCube` object. This could be accomplished by creating an integer array of the specified length, then assigning its values to those obtained by invoking `toss` on the supplied `NumberCube` object. In part (b) students were required to implement the `getLongestRun` method that identifies and returns the starting index of the longest sequence of two or more consecutively repeated values in an array. This involved traversing a supplied array of integer values to locate such sequences.

How well did students perform on this question?

This question appears to have been of average difficulty and was comparable in difficulty to similar questions on previous exams. The mean score was 4.70 out of a possible 9 points. Scores were generally in the 5 to 9 range, but 20 percent of students received scores of 0 or submitted blank papers. Disregarding scores of 0 and blank papers, the mean was 5.88.

What were common student errors or omissions?

Most of the student errors on this question involved incorrect array construction and access, flawed loop-based iteration, and failures to address the question's specifications. A significant number of students were unable to construct an array properly, and there appeared to be confusion of array indices with array elements. There was a proliferation of typical loop boundary errors as well. Solutions included both single and nested loop algorithms, with the nested loop attempts usually implemented incorrectly. Many students also failed to address the question's specifics and did unnecessary work, for example, by generating a random number instead of invoking the required `toss` method, or returning the maximum length rather than the required index. Identifying the maximum length run appeared to be the most difficult task, and solutions often failed when the maximum length run included the end of the array.

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 expect to encounter questions that require interpretation of the textual presentation, a problem-solving skill tested on this exam. They would benefit from extensive practice with word problems that require translation into looping program constructs. Special attention should be given first to the identification of required general behavior and then to the determination of edge conditions, boundary cases, and special initialization or termination constraints. Array and/or `ArrayList` concepts are likely to continue to constitute a substantial portion of future exams, and students need proficiency with their use. As this question demonstrates, common algorithmic tasks, such as finding a maximum value, may be assessed in a variety of contexts.

Question 2

What was the intent of this question?

This question involved reasoning about the code from the GridWorld case study, emphasizing object-oriented concepts. Students demonstrated their understanding of the case study and its interacting classes by extending the `Critter` class to derive a `StockpileCritter` class with modified behavior. This question tested numerous concepts: creating a class, inheriting from an existing class, overriding appropriate methods, and maintaining the overridden methods' postconditions. Students were specifically instructed not to override the `act` method, and they were explicitly cautioned to abide by the postconditions of all methods.

How well did students perform on this question?

This question had the fewest scores of 0 and blank papers by far (less than 13 percent combined), which is extraordinary performance, especially for a case study question. The mean score was 4.64 out of a possible 9 points. Students generally did well, though very few solutions earned a perfect score of 9 because nearly every solution violated a necessary postcondition. Disregarding scores of 0 and blank papers, the mean was 5.32.

What were common student errors or omissions?

The most common student errors were ignoring required solution specifications through either overriding the `act` method or violating well-specified postconditions. For example, a common erroneous approach involved decrementing the stockpile in the `makeMove` method, which violated one of that method's postconditions. There was also apparent confusion over the responsibilities of the various case study methods. Other common errors included declaration of local variables instead of instance variables, incompletely overridden methods, and off-by-one calculations when determining stockpile depletion.

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 reminded to read the problem statements carefully so that they do not overlook important design requirements and implementation constraints. They would also benefit from greater awareness of the principles of good object-oriented design and programming, especially those exhibited in the case study. Key among these are implications of working on larger multideveloper projects, such as the value of appropriate visibility of instance variables and methods, the importance of adhering to solution specifications, and the dangers associated with the violation of encapsulation or postconditions.

Question 3

What was the intent of this question?

This question focused on array traversal, abstraction, and algorithms for accumulation and finding a minimum. Students were provided with the framework of the `BatteryCharger` class that included a private array instance variable with exactly 24 `int` elements, and they were asked to implement two instance methods. The first method, `getChargingCost`, required calculation of a total charging cost given a start time (`startHour`) and a number of hours (`chargeTime`). This could be accomplished by accessing elements of the instance array, beginning with the element at index `startHour`, and traversing in a circular manner (for example, by using the modulus operator), accumulating the values from the array, and returning the sum. The second method, `getChargeStartTime`, required students to return the start time that would allow the battery to be charged at minimal cost. This was best accomplished by invoking the `getChargingCost` method from part (a) for each of the 24 potential start times, comparing the results to determine which achieve the minimum charging cost, and returning that start time.

How well did students perform on this question?

Students performed best on this question out of all the questions on the exam, with more than 40 percent earning scores of 7, 8, or 9. The mean score was 4.75 out of a possible 9 points, even though 25 percent earned scores of 0 or submitted blank papers. Disregarding zeros and blanks, the mean was a very strong 6.35. A somewhat bipolar distribution indicates that this question was straightforward for those who were well prepared, yet quite difficult for those who were not.

What were common student errors or omissions?

Although most students appeared to understand the concept of accumulating, many had difficulty with properly bounding the accumulation process. Most students did not take advantage of the modulus operator (`%`), instead using alternative error-prone approaches. There was also some apparent confusion with 0-based array indexing. A significant number of solutions to part (b) did not use the most straightforward approach of invoking `getChargingCost`. Other common errors included improperly initializing variables, incorrect approaches to finding the minimum, and returning the minimum charging cost rather than the start hour of that minimum charging cost.

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 reminded to read the problem statements carefully and ensure that they are returning the specified result. Here again is an example in which array concepts constitute a substantial portion of the question, so students need to be proficient with their use and with the nature of 0-based indexing. Variable initialization should be addressed in an algorithm-specific context, so that students gain experience with both syntactic and semantic aspects of initialization. More attention to the modulus operator is likely to prove useful, especially for addressing bounded structures by numerical indices. This question again demonstrates that common algorithmic tasks, such as finding a minimum value, may be assessed in a variety of contexts.

Question 4

What was the intent of this question?

This question focused on object-oriented programming, algorithm development, and list processing, requiring significant problem analysis and algorithm design. Students were expected to analyze the problem, design algorithms, and then implement them using the well-specified public interface of a `NumberTile` class. The question involved writing two related methods for a given `TileGame` class. The method `getIndexForFit` determines where a given `NumberTile` object can be added into the `ArrayList` instance variable `board`. The method `insertTile` modifies the state of a `TileGame` object by adding a given `NumberTile` object to `board`. In designing their solutions, students were required to consider the full set of potential cases and to work within the constraints of the public methods provided for `NumberTile`. This question assessed the ability to decompose a stated problem into computational constituents, use algorithmic thinking, rely on abstraction, and demonstrate facility with an `ArrayList` object.

How well did students perform on this question?

This question had the highest number of scores of 0 and blank papers (almost 28 percent). This may indicate students' relative discomfort with more complex problem descriptions and greater emphasis on problem analysis, or it may be because the question required algorithm design rather than simply writing code for a given algorithm. (It was also the last question on the exam, so students could have been tired.) Otherwise, scores were evenly distributed, with a mean score of 4 out of a possible 9 points. Disregarding scores of 0 and blank papers, the mean was 5.54.

What were common student errors or omissions?

Many students seemed unprepared to address this level of problem solving and algorithm development in a free-response question. Incorrect program logic was pervasive, often evidenced by incorrect use of nested if-else constructs and behaviors that resulted in changes of an object's state that were prohibited by the specification. Incorrect use of object-oriented method invocation was common. There was also substantial confusion with working with lists, specifically accessing elements of an `ArrayList` and manipulating `ArrayList` objects.

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 advised to consider the main ideas of a problem statement and take sufficient time to design the solution before turning to code. Emphasis should be placed on determining the required functional behavior (return values) and any required or prohibited side effects (changes to state). Student responses indicated a need to review basic object-oriented programming, such as when it is necessary to invoke an object's methods and how to do so properly. This is another example in which `ArrayList` concepts constitute a substantial portion of the question; it is a reminder that students need proficiency with the manipulation of `ArrayList` objects and practice with accessing their elements.