2023

AP® Computer Science A

Sample Student Responses and Scoring Commentary

Inside:

Free-Response Question 3

☑ Scoring Guidelines

☑ Student Samples

☑ Scoring Commentary

Applying the Scoring Criteria

Apply the question scoring criteria first, which always takes precedence. Penalty points can only be deducted in a part of the question that has earned credit via the question rubric. No part of a question (a, b, c) may have a negative point total. A given penalty can be assessed only once for a question, even if it occurs multiple times or in multiple parts of that question. A maximum of 3 penalty points may be assessed per question.

1-Point Penalty

v) Array/collection access confusion (`[] get`)
w) Extraneous code that causes side-effect (e.g., printing to output, incorrect precondition check)
x) Local variables used but none declared
y) Destruction of persistent data (e.g., changing value referenced by parameter)
z) Void method or constructor that returns a value

No Penalty

- Extraneous code with no side-effect (e.g., valid precondition check, no-op)
- Spelling/case discrepancies where there is no ambiguity*
- Local variable not declared provided other variables are declared in some part
- `private` or `public` qualifier on a local variable
- Missing `public` qualifier on class or constructor header
- Keyword used as an identifier
- Common mathematical symbols used for operators (× • ÷ ≤ ≥ <> ≠)
- `[]` vs. `()` vs. `<>`
- `=` instead of `==` and vice versa
- `length/size` confusion for array, `String`, `List`, or `ArrayList`; with or without `( )`
- Extraneous `[]` when referencing entire array
- `[i,j]` instead of `[i][j]`
- Extraneous size in array declaration, e.g., `int[size] nums = new int[size];`
- Missing `;` where structure clearly conveys intent
- Missing `{ }` where indentation clearly conveys intent
- Missing `( )` on parameter-less method or constructor invocations
- Missing `( )` around `if` or `while` conditions

Spelling and case discrepancies for identifiers fall under the "No Penalty" category only if the correction can be* ***unambiguously* *inferred from context, for example, "ArayList" instead of "ArrayList". As a counterexample, note that if the code declares* `"int G=99, g=0;"`*, then uses* `"while (G < 10)"` *instead of* `"while (g < 10)"`*, the context does* ***not*** *allow for the reader to assume the use of the lower case variable.*

Question 3: Array / ArrayList — 9 points

Canonical solution

(a) — **4 points**

```
public void cleanData(double lower, double upper)
{
   for (int i = temperatures.size() - 1; i >= 0; i--)
   {
      double temp = temperatures.get(i);
      if (temp < lower || temp > upper)
      {
         temperatures.remove(i);
      }
   }
}
```

(b) — **5 points**

```
public int longestHeatWave(double threshold)
{
   int waveLength = 0;
   int maxWaveLength = 0;

   for (double temp : temperatures)
   {
      if (temp > threshold)
      {
         waveLength++;
         if (waveLength > maxWaveLength)
         {
            maxWaveLength = waveLength;
         }
      }
      else
      {
         waveLength = 0;
      }
   }
   return maxWaveLength;
}
```

(a) `cleanData`

	Scoring Criteria	Decision Rules	
1	Traverses `temperatures` (*no bounds errors*)	Responses **can** still earn the point even if they • do a forward traversal of the list • skip a value because removal from the list is handled incorrectly • use an enhanced `for` loop, as long as the list element is used in the body of the loop Responses **will not** earn the point if they • fail to ever access an element of `temperatures` in the loop • access the elements of `temperatures` incorrectly	**1 point**
2	Determines whether an element of temperature list should be removed, using `lower` and `upper`	Responses **can** still earn the point even if they • access elements of temperature list incorrectly Responses **will not** earn the point if they • apply incorrect comparison (`<` vs `<=`) or logic (`||` vs `&&`) to identify elements of list for removal	**1 point**
3	Calls `remove` (or equivalent) on temperature list with an appropriate parameter	Responses **can** still earn the point even if they • add the element to a new `ArrayList` that is not declared, is declared incorrectly, or is not assigned to the instance variable, as long as the order of elements is maintained Responses **will not** earn the point if they • call `remove` or `add` incorrectly	**1 point**
4	Removes all and only identified elements of temperature list (*algorithm*)	Responses **can** still earn the point even if they • call `remove` incorrectly • access the elements of temperature list incorrectly Responses **will not** earn the point if they • add elements to a new `ArrayList` that is not declared, is declared incorrectly, or is not assigned to the instance variable • skip a temperature list element in the traversal because removal is not handled correctly	**1 point**
		Total for part (a)	**4 points**

(b) `longestHeatWave`

	Scoring Criteria	Decision Rules	
5	Traverses `temperatures` (*no bounds errors*)	Responses **will not** earn the point if they • fail to access an element of `temperatures` in the loop • access the elements of `temperatures` incorrectly	**1 point**
6	Compares an element of temperature list to `threshold` (*in the context of a loop*)	Responses **can** still earn the point even if they • always compare `threshold` to the same list element Responses **will not** earn the point if they • apply incorrect comparison (`>` vs `>=`) to identify heat wave days	**1 point**
7	Initializes and increments the length of a heat wave (*in the context of a loop or condition*)	Responses **can** still earn the point even if they • fail to reset the length of the current heat wave when the heat wave ends	**1 point**
8	Determines the length of at least one heat wave (*algorithm*)	Responses **will not** earn the point if they • fail to reset the length of the current heat wave when the heat wave ends	**1 point**
9	Identifies the longest heat wave and returns its length (*algorithm*)		**1 point**
		Total for part (b)	**5 points**
	Question-specific penalties		
	None		
		Total for question 3	**9 points**

Q3 Sample A 1 of 2

● **Important:** Completely fill in the circle that corresponds to the question you are answering on this page.

Question 1 ○ Question 2 ○ [✓ Question 3 ● Part A] Question 4 ○

Begin your response to each question at the top of a new page.

```
public void cleanData(double lower, double upper){
   for (int i=0; i<temperatures.size(); i++){
      if ((temperatures.get(i)<lower)||(temperatures.get(i)
         >upper)){
         temperatures.remove(i);
         i--;
      }
    }
  }
```

Use a pencil only. Do NOT write your name. Do NOT write outside the box.

0056149

Q3162/5

Important: Completely fill in the circle that corresponds to the question you are answering on this page.

Question 1 ○ Question 2 ○ [✓ Question 3] Question 4 ○

Part b

Begin your response to each question at the top of a new page.

```
public int longestHeatWave(double threshold) {
    int count = 0; int temp = 0;
    for (int i=0; i<temperatures.length; i++) {
      if (temperatures.get(i) > threshold) {
        temp = 1;
        while (temperatures.get(i) >threshold) {
          temp++;
        }
      }
      if (count<temp) {
        count = temp;
      }
    }
    return count;
}
```

Use a pencil only. Do NOT write your name. Do NOT write outside the box.

Q3 Sample B 1 of 2

Important: Completely fill in the circle that corresponds to the question you are answering on this page.

Question 1 ○ Question 2 ○ [✓ Question 3] Question 4 ○

Begin your response to each question at the top of a new page.

```
(A) public void cleanData(double lower, double upper){

  for(int i=0; i < temperatures.length; i++) {
   if(temperatures.get(i) < lower || temperatures.get(i) > upper){
     temperatures.remove(temperatures.get(i));
   }
  }
 }
```

Q3 Sample B 2 of 2

Important: Completely fill in the circle that corresponds to the question you are answering on this page.

Question 1 ○ Question 2 ○ [✓ Question 3 ●] Question 4 ○

Begin your response to each question at the top of a new page.

```
(b) public int longestHeatWave(int threshold) {
        int count=0;
        for(int i=0; i< temperatures.length; i++) {

          if(temperatures.get(i) > threshold &&
               temperatures.get(i-1) < threshold &&
               temperatures.get(i+1) > threshold) {
               count++;

          }

        } return count;

      }
```

Use a pencil only. Do NOT write your name. Do NOT write outside the box.

Part A:

```
Public void cleanData(double lower, double upper) {
        for(int i=0; i< temperatures.length; i++) {
            if (temperatures.get(i) < lower) {
                temperatures.remove(i);
            }
            else if (temperatures.get(i) > upper) {
                temperatures.remove(i)
            }
        }
        return temperatures;
    }
```

Part B:

```
Public int longestHeatWave(double threshold) {
    double temp = 0;
    for(int i=0; i< temperatures.length; i++) {
        if ( temperatures.get(i) && temperatures.get(i+1) > threshold);
        {
            double temp = temperatures.get(i+1) - temperatures.get(i);
            double heatWave = temp;
            if (temp > heatWave) {
                return temp;
            }
            else {
            return heatWave;
            }
```

Question 3

Note: Student samples are quoted verbatim and may contain spelling and grammatical errors.

Overview

This question tested the student's ability to:

- Write program code to create objects of a class and call methods.
- Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.
- Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` objects.

This question involved the traversal, analysis, and manipulation of an `ArrayList` object containing numbers modeling recorded temperatures. Students were expected to write two methods of the `WeatherData` class, using its `ArrayList` instance variable.

In part (a) students were expected to write a loop that accessed each element of the `ArrayList` instance variable `temperatures` and removed any elements that were outside of a range specified by the parameters `upper` and `lower`. Inside the loop, students were expected to use a conditional to identify elements for removal from the list and to call the `remove` method on the instance variable `temperatures`.

In part (b) students were asked to (1) identify the lengths of any heat waves of consecutive elements in `temperatures` that were greater than the parameter `threshold`, and (2) determine and return the length of the longest of the identified heat waves.

Sample: 3A
Score: 8

In part (a) point 1 was earned because the response iterates over all elements of the list and correctly accesses each one. This response uses a `for` loop to iterate forward through the list, but other correct approaches exist. Point 2 was earned because the response makes the correct comparisons between an element of `temperatures` and the parameters `upper` and `lower`. To earn the point, the response must use the correct comparison operator (e.g., `<` or `>`) and the correct logic operator (e.g., `||`). Point 3 was earned because the response correctly calls the `remove` method to remove an element from the `temperatures` list. Point 4 was earned because the algorithm results in the correct values being removed from `temperatures`. When a response uses an increasing index in a `for` loop, a correct algorithm requires a decrement of the loop control variable after the call to `remove` in order to avoid skipping elements.

In part (b) point 5 was earned because the response iterates over all elements of `temperatures`, accessing each one properly. This response uses a `for` loop with an incremented index value, but other approaches could also be correct. Note that the use of `length` instead of `size` in the `for` loop is one of the minor errors for which no penalty is assessed on this exam. (See the "No Penalty"

Question 3 (continued)

section on page 1 of the Scoring Guidelines for a complete list.) Point 6 was earned because elements of the list are appropriately compared to `threshold`. To earn the point, the response must use the correct comparison operator. Most correct responses will not require the use of any logical operators. Point 7 was earned because the variable `temp`, which measures the length of a heat wave, is initialized and then incremented in some cases, based on the results of a comparison involving an element of `temperatures`. Point 8 was not earned because the loop control variable `i` is not updated in the `while` loop, resulting in an infinite loop. Point 8 assesses the algorithm implemented to identify a single heat wave. As a result of the infinite loop, the length of a heat wave is not correctly determined because the end of a heat wave is never identified. Point 9 was earned because the algorithm compares the identified current heat wave length, `temp`, to a maximum heat wave length, `count`; updates the maximum length when appropriate; and eventually returns the maximum length. Note that the infinite loop causes point 8 to not be earned and is not assessed as part of point 9.

Sample: 3B
Score: 4

In part (a) point 1 was earned because the response correctly accesses all elements of `temperatures`. Using `length` instead of `size` is one of the minor errors for which no penalty is assessed on this exam. (See the "No Penalty" section on page 1 of the Scoring Guidelines for a complete list.) Point 2 was earned because the response makes the correct comparisons between an element of `temperatures` and the parameters `upper` and `lower`. Point 3 was earned because the response correctly calls the `remove` method to remove an element from the `temperatures` list. The argument to the `remove` method in this response is the element to be removed, rather than the index of the element to be removed. This is valid and removes the first instance of that element in the list. Point 4 was not earned because the algorithm skips over any element immediately following a removed element. Decrementing the variable `i` after the call to `remove` would have avoided this error.

In part (b) point 5 was not earned because the response attempts to access the list element at index `i + 1`, causing an out-of-bounds error at the end of the list. A similar error occurs when the response attempts to access the list element at index `i - 1` at the beginning of the list. Point 6 was not earned because the list element at index `i - 1` is incorrectly compared to `threshold`. When multiple comparisons to `threshold` are made, all must be correct to earn the point. A correct comparison must either test if the element is strictly greater than `threshold` or if the element is less than or equal to `threshold`. Point 7 was earned because the variable `count`, which measures the length of a heat wave, is properly initialized and incremented as a result of a comparison involving `temperatures`. Point 8 was not earned because the variable `count` is never reset to zero, so the length of a given heat wave is not determined. Furthermore, the response does not count all temperatures above the threshold due to the additional tests for temperatures at indices `i - 1` and `i + 1`. Point 9 was not earned because no algorithm to find the length of the longest heat wave is implemented.

Question 3 (continued)

Sample: 3C
Score: 3

In part (a) point 1 was earned because the response accesses all elements of `temperatures`. Point 2 was earned because the response makes the correct comparisons between an element of `temperatures` and the parameters `upper` and `lower`. Point 3 was earned because the response correctly calls the `remove` method to remove an element from the `temperatures` list. Point 4 was not earned for either of the following reasons. First, the algorithm skips over any element immediately following a removed element. Decrementing the variable `i` after the call to `remove` would have avoided this error. Second, the response returns the instance variable `temperatures` from within the `void` method. Returning or printing the list makes the algorithm incorrect and is assessed in this point. Note that there is a related penalty in the "1-Point Penalty" list, but that penalty is not applied because the error is assessed as part of the algorithm within this rubric point.

In part (b) point 5 was not earned because the response attempts to access the element at index `i + 1`, causing an out-of-bounds error at the end of the list. Point 6 was not earned because the `boolean` expression is invalid; there is no comparison involving the first `get` operation. When multiple comparisons to `threshold` are made, all must be correct in order to earn the point. Point 7 was not earned because no variable measuring the length of a heat wave is ever incremented. Point 8 was not earned because no variable measuring the length of a heat wave is ever reset at the end of an identified heat wave. The response finds the difference between consecutive temperature values rather than incrementing a count, and therefore, it has not determined the length of at least one heat wave. Point 9 was not earned because no variable is updated when a longer heat wave is identified. Furthermore, the `return` statements in both the `if` and `else` blocks result in an early return in the first iteration of the loop.