Chief Reader Report on Student Responses:

2018 AP® Computer Science A Free-Response Questions

- Number of Students Scored 65,133
- Number of Readers 317
- Score Distribution

Exam Score	N	%At
5	16,105	24.7
4	13,802	21.2
3	14,222	21.8
2	7,738	11.9
1	13,266	20.4

- Global Mean 3.18

The following comments on the 2018 free-response questions for AP® Computer Science A were written by the Chief Reader, John Cigas, Professor, Park 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 preparation 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

Task: Methods and Control
Topic: Frog Simulation
Max. Points: 9
Mean Score: 5.34

What were the responses to this question expected to demonstrate?

This question tested the student's ability to:

- *Write program code to create objects of a class and call methods; and*
- *Write program code to satisfy methods using expressions, conditional statements, and iterative statements.*

Students were provided with the specifications of the `FrogSimulation` class. The `FrogSimulation` class encapsulates a simulation of a frog hopping in a straight line. It contains two private integer instance variables, `goalDistance` and `maxHops`, which represent the distance in inches from the starting point to the goal and the maximum number of hops allowed to reach the goal. It also contains a private method, `hopDistance`, which returns an integer representing the distance in inches to be moved when the frog hops. Implementation for this method was not shown.

In part (a), students were asked to write the `FrogSimulation` method `simulate`, which determines whether a frog is successful in reaching `goalDistance`. Students were required to use the private method `hopDistance` within the context of a loop to update an initialized variable representing the frog's position. The loop iterates until one of the following conditions becomes true.

- The frog has reached or passed the goal, in which case a value of `true` is immediately returned
- The frog has reached a negative position, in which case a value of `false` is immediately returned
- A frog has taken `maxHops` hops without reaching the goal, in which case a value of `false` is returned

In part (b), students were asked to write the `FrogSimulation` method `runSimulations(int num)`, which uses a loop to call the `simulate` method `num` times. Each time `simulate` returns `true`, a previously initialized variable is incremented. The method returns a decimal value representing the proportion of simulations in which the frog successfully reached or passed the goal.

How well did the response address the course content related to this question? How well did the responses integrate the skills required on this question?

Write program code to create objects of a class and call methods.

Both parts of this question involved calling a method within the context of a loop and then using the returned result. Most responses were successful in calling the methods `hopDistance` and `simulate` within their respective loops and then using the returned result appropriately.

Write program code to satisfy methods using expressions, conditional statements, and iterative statements.

Both parts of this question involved the use of a loop with a specific upper bound. The majority of responses demonstrated this concept. However, in part (a), the question included additional conditions to trigger an early termination of the loop. Responses were less successful with this concept, either using an incorrect comparison operator, placing the required conditional statements outside of the loop, or omitting at least one of the required conditions completely.

In part (b), most responses calculated a proportion by finding the quotient of two values. However, a significant number of responses failed to return a correctly calculated decimal value.

What common student misconceptions or gaps in knowledge were seen in the responses to this question?

Common Misconceptions/Knowledge Gaps of: *Write program code to create objects of a class and call methods.*	*Responses that Demonstrate Understanding*

Responses failed to call the `FrogSimulation` instance method `hopDistance` from within the `FrogSimulation` class correctly.
 	
`position = sim.hopDistance();`
`position = FrogSimulation.hopDistance();`
`position = obj.hopDistance();`
	`position = hopDistance();`
	`position = this.hopDistance();`

Common Misconceptions/Knowledge Gaps *Write program code to satisfy methods using expressions, conditional statements, and iterative statements.*	*Responses that Demonstrate Understanding of:*

Responses failed to check for a negative position.
 	
`int position = 0;`
`for (int x = 0; x < maxHops; x++)`
`{`
`  postion += hopDistance();`
`  if (position >= goalDistance)`
`  {`
`  return true;`
`  }`
`}`
`return false;`
 	
or
	`int position = 0;`
	`for (int x = 0; x < maxHops; x++)`
	`{`
	`  position += hopDistance();`
	`  if (position >= goalDistance)`
	`  {`
	`  return true;`
	`  }`
	`  if (position < 0)`
	`  {`
	`  return false;`
	`  }`
	`}`
	`return false;`
	
	or

```
int position = 0;
for (int x = 0; x < maxHops; x++)
{
   position += hopDistance();
}
if (position >= goalDistance)
{
   return true;
}
else
{
   return false;
}
```

or

```
int position = 0;
int numHops = 0;
while (position < goalDistance && numHops < maxHops)
{
   position += hopDistance();
   numHops++;
}
return position >= goalDistance;
```

```
int position = 0;
int numHops = 0;
while (position < goalDistance && position >= 0 &&
       numHops < maxHops)
{
   position += hopDistance();
   numHops++;
}
return position >= goalDistance;
```

Responses used integer arithmetic instead of `double` arithmetic to calculate a decimal value.

```
int count = 0;
for (int x = 0; x < num; x++)
{
   if (simulate())
   {
      count++;
   }
}
return count / num;
```

Declare a variable as `double` for use in the calculation

```
double count = 0.0;
for (int x = 0; x < num; x++)
{
   if (simulate())
   {
      count++;
   }
}
return count / num;
```

Responses used improper casting to produce a `double` quotient from two integer values.
 	
`int count = 0;`
`for (int x = 0; x < num; x++)`
`{`
`  if (simulate())`
`  {`
`  count++;`
`  }`
`}`
`return (double)(count / num);`
	Cast an integer variable as a `double` within the calculation
	
	`int count = 0;`
	`for (int x = 0; x < num; x++)`
	`{`
	`  if (simulate())`
	`  {`
	`  count++;`
	`  }`
	`}`
	`return (double)count / num;`

Responses looped an incorrect number of times.
 	
`for (int x = 0; x <= num; x++)`
`{`
`  if (simulate())`
`  {`
`  count++;`
`  }`
`}`
	`for (int x = 0; x < num; x++)`
	`{`
	`  if (simulate())`
	`  {`
	`  count++;`
	`  }`
	`}`

Responses included an infinite loop.
 	
`int x = 0;`
`int count = 0;`
`while (x < num)`
`{`
`  if (simulate())`
`  {`
`  count++;`
`  }`
`}`
	`int x = 0;`
	`int count = 0;`
	`while (x < num)`
	`{`
	`  if (simulate())`
	`  {`
	`  count++;`
	`  }`
	`  x++;`
	`}`

Based on your experience at the AP® Reading with student responses, what advice would you offer to teachers to help them improve the student performance on the exam?

Write program code to create objects of a class or call methods.

- Students need to practice invoking different kinds of methods.
    - Assign problems requiring students to invoke methods that they don't implement.
    - Provide students with a partially developed class containing only private helper methods. Require students to develop public methods that call the private methods. Emphasize the fact that the private methods do not need to be called on an instance of the class.

Write program code to satisfy methods using expressions, conditional statements, and iterative statements.

- Students need to know the difference between integer and `double` arithmetic.
    - Assign problems that incorporate common formulas containing integer fractions to calculate a result so that students can see the difference between
      ```
      double areaTriangle = (1 / 2) * base * height;
      ```
      and
      ```
      double areaTriangle = (1.0 / 2) * base * height;
      ```
    - Assign problems that use integer values to calculate `double` results, for example, compute the average age of students in the class.
- Students need to determine loop bounds correctly.
    - Assign problems that require students to count something a specific number of times.
    - Assign problems involving loops with varying lower and upper bounds (perhaps based upon user input). Be sure that students note and confirm the correct number of loop iterations.
- Students need to practice identifying multiple loop termination conditions.
    - Create problems that require students to implement loops with multiple terminating conditions. Sequential search is one such example.
    - Provide students with a prewritten loop containing conditionals that terminate the loop by invoking the break statement. Have the students rewrite the code to produce the same result without invoking break.

What resources would you recommend to teachers to better prepare their students for the content and skill(s) required on this question?

Suggested resources include:

Write program code to create objects of a class and call methods.

- The 2017 free-response question number 3, PhraseEditor, requires students to call methods of the newly presented class. This provides students with practice calling methods that were not studied in class. This resource can be found here: https://apcentral.collegeboard.org/courses/ap-computer-science-a/exam?course=ap-computer-science-a
- The Practice-It! website hosted by the University of Washington offers practice for students to analyze program code that calls methods. This practice is consolidated into Chapter 3. This resource can be found here: https://practiceit.cs.washington.edu/problem/list

Write program code to satisfy methods using expressions, conditional statements, and iterative statements.

- The Runestone Interactive Java Review offers an interactive environment for students to practice course content. Specifically, the Java Basics: Classes and Objects section and the Object-Oriented Concepts section would be most helpful for this question. This resource can be found here: http://interactivepython.org/runestone/static/JavaReview/index.html

Question #2

Task: `ArrayList` Processing
Topic: Word Pair
Max. Points: 9
Mean Score: 4.38

What were the responses to this question expected to demonstrate?

This question tested the student's ability to:

- *Write program code to define a new type by creating a class; and*
- *Write program code to create objects of a class and call methods; and*
- *Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` objects.*

Students were asked to write a constructor and a method of the `WordPairList` class. In writing the constructor, students were expected to access an array of strings in order to populate an `ArrayList` of `WordPair` objects. Students were also expected to traverse the list of objects in order to count how many elements met a specified requirement. A provided `WordPair` class is used to represent pairs of words extracted from the array.

In part (a), the students were asked to write a constructor for the `WordPairList` class. Students needed to recognize that the `ArrayList` instance variable must be constructed before elements can be added. To populate the list, the students were expected to write a loop structure to pair each element from the `words` array parameter with each of the subsequent elements in the array. Students were expected to construct a `WordPair` object from each of the paired elements and add each `WordPair` object to the `allPairs` instance variable.

In part (b), students were expected to access all the elements of `allPairs` to count how many `WordPair` elements consisted of pairs of matching strings. Students were expected to call the `WordPair` methods `getFirst` and `getSecond` to access each word component in the pair. To compare the words, students were expected to use appropriate methods of the `String` class, such as `equals` or `compareTo`. To count the number of matching words, students were expected to declare and initialize an accumulator before their loop structure and increment it only when a match was found. While the method was required to return a value, the return of the accumulator was not assessed in this question.

How well did the response address the course content related to this question? How well did the responses integrate the skills required on this question?

Write program code to define a new type by creating a class.

> Although most responses added items to an `ArrayList`, very few responses demonstrated instantiating an instance variable in the constructor, and of those that attempted this, many did the instantiation incorrectly.

Write program code to create objects of a class and call methods.

> While most responses attempted to create a `WordPair` object, many implemented this incorrectly. Far more challenging was writing nested loops to pair each element from an array with each subsequent element. Some responses did this correctly, but other responses erred in not preventing the pairing of an array element with the preceding elements or itself.

Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` objects.

> Responses demonstrated the ability to count how many elements of a list met a condition. The responses that successfully accessed a list element were able to use the object to access the required components of the pair. Most responses also demonstrated the ability to correctly compare strings.

What common student misconceptions or gaps in knowledge were seen in the responses to this question?

Common Misconceptions/Knowledge Gaps of: *Write program code to define a new type by creating a class.*	*Responses that Demonstrate Understanding*
Most responses failed to instantiate the private instance variable `allPairs`. Some responses constructed and assigned a new `ArrayList` to a local variable named `allPairs` instead of the instance variable `allPairs`. `ArrayList<WordPair> allPairs = new ArrayList<WordPair>();`	`allPairs = new ArrayList<WordPair>();`

Write program code to create objects of a class and call methods.

Instantiating the instance variable in part (a) proved to be a significant challenge. The responses demonstrated that the use of the keyword `new` in the instantiation of objects is a frequent omission.

Common Misconceptions/Knowledge Gaps of: *Write program code to create objects of a class and call methods.*	*Responses that Demonstrate Understanding*

Some responses failed to use the keyword `new` when constructing a new `WordPair`.
 	
`allPairs.add(WordPair(words[j], words[k]));`
 	
Some responses failed to construct a new `WordPair`.
 	
`allPairs.add(words[j], words[k]);`
	`allPairs.add(new WordPair(words[j], words[k]));`

Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` objects.

Most of the errors in implementing a nested loop involved either not accessing the last element of the array or exceeding the bounds of the array.

Common Misconceptions/Knowledge Gaps of: *Write program code to create, traverse, and manipulate elements in 1D array or* `ArrayList` *objects.*	*Responses that Demonstrate Understanding*

Some responses tried to treat the `ArrayList allPairs` like an array or tried to use `ArrayList` methods incorrectly when adding to the list.
 	
`allPairs[j] = pair;`
`allPairs.get(j) = pair;`
`allPairs.set(j, pair);`
	`allPairs.add(pair);`

Some responses only added consecutive pairs of elements to `allPairs`.
 	
`for (int j = 0; j < words.length - 1; j++)`
`{`
`  allPairs.add(new WordPair(words[j], words[j + 1]);`
`}`
	`for (int j = 0; j < words.length - 1; j++)`
	`{`
	`  for (int k = j + 1; k < words.length; k++)`
	`  {`
	`  allPairs.add(new WordPair(words[j], words[k]));`
	`  }`
	`}`

Some responses added all necessary pairs and all unnecessary, self-to-self pairs to `allPairs`.
 	
`for (int j = 0; j < words.length - 1; j++)`
`{`
`  for (int k = j; k < words.length; k++)`
`  {`
`  WordPair pair = new WordPair(words[j], words[k]);`
`  allPairs.add(pair);`
`  }`
`}`
	`for (int j = 0; j < words.length; j++)`
	`{`
	`  for (int k = j + 1; k < words.length; k++)`
	`  {`
	`  WordPair pair = new WordPair(words[j], words[k]);`
	`  allPairs.add(pair);`
	`  }`
	`}`

Some responses added all pairs to `allPairs`.
 	
`for (int j = 0; j < words.length - 1; j++)`
`{`
`  for (int k = 0; k < words.length; k++)`
`  {`
`  WordPair pair = new WordPair(words[j], words[k]);`
`  allPairs.add(pair);`
`  }`
`}`
	`for (int j = 0; j < words.length; j++)`
	`{`
	`  for (int k = 0; k < words.length; k++)`
	`  {`
	`  if (j < k)`
	`  {`
	`  WordPair pair = new WordPair(words[j],words[k]);`
	`  allPairs.add(pair);`
	`  }`
	`  }`
	`}`

Some responses used array notation when accessing elements of `allPairs`.
 	
`allPairs[i].getFirst()`
 	
Some responses forgot to access the list when accessing `allPairs` or improperly used a parameter in the `getFirst` or `getSecond` methods
 	
`allPairs.getFirst()`
`allPairs.getFirst(i)`
	`allPairs.get(i).getFirst()`

When comparing strings, some responses incorrectly compared the strings using the `==` operator.
 	
`pair.getFirst() == pair.getSecond()`
	`pair.getFirst().equals(pair.getSecond())`
	
	or
	
	`pair.getFirst().compareTo(pair.getSecond()) == 0`

Some responses were unable to determine how to retrieve the components of a `WordPair` object.
 	
`pair[0]` and `pair[1]`
 	
`pair.first()` and `pair.second()`
	`pair.getFirst()` and `pair.getSecond()`

A few responses omitted the initialization of the accumulator. `int count;`	`int count = 0;`

Based on your experience at the AP® Reading with student responses, what advice would you offer to teachers to help them improve the student performance on the exam?

Write program code to define a new type by creating a class.

- Students need to know how to write constructors for objects that do more than simply assign parameters to instance variables.
    - Have students create a collection class that collects some object of interest to them.
    - Have students create a constructor that is complex and requires more than just assigning values to the instance variables, such as using an array or list parameter to construct and populate an instance variable.
    - Reinforce that if an instance variable is an object, such as an array or list, it needs to be instantiated in a constructor. Also, emphasize that constructors do not return values and have no return type.

Write program code to create objects of a class and call methods.

- Students need to know how to create and use objects of types unknown to them.
    - Have students use many different classes to construct objects.
    - Develop solutions requiring students to call methods when only given the method signature.

Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` *objects.*

- Students need to practice traversing both arrays and lists while computing some result.
    - Have students compute the count or sum of a variety of different types of objects that are stored in an array or a list.
    - Have students identify elements that meet a specified condition using an object's accessor methods.
    - Have students search for matches using the appropriate comparison for both object and primitive data.
- Students need to practice determining the proper lower bound and upper bound for loops.
    - Assign problems that have loops requiring different lower (do not start at zero) and/or different upper bounds (different than < length).
    - Assign problems that have nested loops where the loop control variable of the inner loop is dependent on the loop control variable of the outer loop.
- Students need to learn to evaluate their code to find mistakes.
    - Give students code samples and have them trace the code to find the mistake.
    - Teach students to check that the use of their loop control variable is consistent throughout each loop header and body, especially when writing nested loops.
    - After creating a loop structure, have students check that the bounds are correct and all variables are initialized correctly.
    - After completing the implementation of a method, encourage students to re-read the comment above the method heading, checking to see that all requirements have been met.

What resources would you recommend to teachers to better prepare their students for the content and skill(s) required on this question?

Suggested resources include:

Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` *objects.*

- Codingbat.com provides students with practice writing the body of a method based on a given specification. The website provides several Array problems for students to practice.

- The Runestone Interactive Java Review offers an interactive environment for students to practice course content. Specifically, the List and ArrayList section would be most helpful for this question. This resource can be found here: http://interactivepython.org/runestone/static/JavaReview/index.html

Question #3

Task: Class Design
Topic: Code Word Checker
Max. Points: 9
Mean Score: 5.11

What were the responses to this question expected to demonstrate?

This question tested the student's ability to:

- *Write program code to define a new type by creating a class; and*
- *Write program code to create objects of a class and call methods; and*
- *Write program code to satisfy methods using expressions, conditional statements, and iterative statements.*

Students were asked to design the class `CodeWordChecker` so that it implements the method `isValid` in the given interface, `StringChecker`. In addition to demonstrating an understanding of class constructors and method header syntax, students had to correctly declare, initialize, access, and update instance variables. Students were expected to properly encapsulate their data members by declaring them as `private` and to properly define the interface method `isValid` and expose it by declaring the method `public`. Students also had to recognize that two constructors needed to be included in order to conform to the examples given.

This question primarily addresses the fundamental skill of how to create a class so that it adheres to a specific interface. However, there is also a logic component required to make the class operational. The students had to demonstrate an understanding of how to control the visibility of class elements so that the language would enforce the separation. The internal representation had to be hidden and inaccessible from the client and the interface method had to be defined and made publicly available.

Implementing the `isValid` method required students to find the length of a string and check if it was within a specified range and also to determine if one string was a substring of another. Students then needed to return an appropriate Boolean value based on the result of their logic.

How well did the response address the course content related to this question? How well did the responses integrate the skills required on this question?

Overall, students who understood the concept of making a class match the specifications scored highly on this question, though not all students understood the concept.

Write program code to define a new type by creating a class.

> Responses were generally able to set up the class structure with instance variables. Some lost these points for omitting keywords like `class` or `private`. Some responses combined the constructor heading with the class heading but this occurred less frequently than in the previous year. However, many responses included the code for `isValid` as part of one or both constructors.

Write program code to create objects of a class and call methods; and
Write program code to satisfy methods using expressions, conditional statements, and iterative statements.

> Most responses were able to write a bounds check on the length of a string. They called a method to get the length and they used some sort of `and` logic to check both sides of the range. However, many times they still lost the point because their bounds were not exact or they put the check inside a constructor instead of inside the `isValid` method.

Most responses checked for the presence of the disallowed string in some way. Those that tried to implement a contains-like loop, instead of calling a string method like `indexOf` or `contains`, failed because the loop stopped one position too early.

What common student misconceptions or gaps in knowledge were seen in the responses to this question?

Common Misconceptions/Knowledge Gaps of: *Write program code to define a new type by creating a class.*	*Responses that Demonstrate Understanding*

Responses omitted the keyword `class` or the entire `implements` clause.
 	
`public CodeWordChecker implements StringChecker`
 	
`public class CodeWordChecker`
 	
Responses used `extends` instead of `implements`.
 	
`public class CodeWordChecker extends StringChecker`
 	
Responses included parameters in the class header. This frequently coincided with missing or incorrect instance variables.
 	
`public class CodeWordChecker`
`  implements StringChecker(int min,`
`  int max, String str)`
	`public class CodeWordChecker implements StringChecker`
	
	The public access modifier was not strictly required.
	
	`class CodeWordChecker implements StringChecker`

Failure to declare *any* instance variables meant that students could not earn points for the declarations, the initialization in the 3-parameter constructor, the initialization in the 1-parameter constructor, the checking of the length of the string parameter, or the checking for the unwanted string.
 	
Some responses that declared instance variables omitted the `private` access specifier.
 	
`int minLength;`
`int maxLength;`
`String notAllowed;`
	`private int minLength;`
	`private int maxLength;`
	`private String notAllowed;`

Responses frequently implemented the 3-parameter constructor but then omitted the 1-parameter constructor.

Responses sometimes reversed the assignments to the instance variables.

```
public CodeWordChecker(int minLen,
       int maxLen, String symbol)
{
   minLen = minLength;
   maxLen = maxLength;
   symbol = notAllowed;
}
```

Responses occasionally changed the order of the parameters in the 3-parameter constructor.

```
public CodeWordChecker(int maxLen, int minLen, String str)
{
   minLength = minLen;
   maxLength = maxLen;
   notAllowed = str;
}
```

Responses sometimes created local variables in the constructor and assigned the parameters to these instead of the instance variables.

```
public CodeWordChecker(int minLen, int maxLen,
       String symbol)
{
   int minLength = minLen;
   int maxLength = maxLen;
   String notAllowed = symbol;
}
```

```
public CodeWordChecker(int minLen,
      int maxLen, String symbol)
{
   minLength = minLen;
   maxLength = maxLen;
   notAllowed = symbol;
}


public CodeWordChecker(String symbol)
{
   minLength = 6;
   maxLength = 20;
   notAllowed = symbol;
}
```

The `public` access modifier on the constructor header is not strictly required.

```
CodeWordChecker(int minLen, int maxLen, String symbol)
CodeWordChecker(String symbol)
```

Instance variables can be initialized at declaration, thus not requiring the assignment in the 1-parameter constructor.

```
private int minLength = 6;
private int maxLength = 20;
private String notAllowed;

public CodeWordChecker(String symbol)
{
   notAllowed = symbol;
}
```

The 1-parameter constructor calls the 3-parameter constructor.

```
public CodeWordChecker(String symbol)
{
   this(6, 20, symbol)
}
```

Responses frequently omitted the `isValid` method from the `StringChecker` interface.

Many responses omitted the `isValid` method, but attempted to write the `isValid` code inside a constructor.

```
public CodeWordChecker(int minLen, int maxLen, String str)
{
   minLength = minLen;
   maxLength = maxLen;
   notAllowed = str;
   if (str.length() < min ||
       str.length() > max ||
       str.indexOf(notAllowed) != -1)
   {
      return true;
   }
   else
   {
      return false;
   }
}
```

```
public boolean isValid(String str)
{
   return str.length() >= minLength &&
          str.length() <= maxLength &&
          str.indexOf(notAllowed) == -1;
}
```

Responses omitted the `public` access modifier on the `isValid` method header. Ocassionally, students omitted the `boolean` return type or the parameter.

```
boolean isValid(String str)
public isValid(String str)
public boolean isValid()
```

Some responses added additional parameters to the method heading.

```
public boolean isValid(int min, int max, String str)
```

```
public boolean isValid(String str)
```

Common Misconceptions/Knowledge Gaps of: *Write program code to create objects of a class and call methods.*	*Responses that Demonstrate Understanding*

Many responses used an incorrect comparison when searching for the unwanted string.
`str.indexOf(notAllowed) <= 0`
 	
Responses often reversed the variables when searching for the unwanted string.
`notAllowed.indexOf(str)`
 	
Responses frequently used `equals` or `compareTo` when testing for the unwanted string.
`str.equals(notAllowed)`
`str.compareTo(notAllowed) == 0`
	`str.indexOf(notAllowed) == -1`
	
	or
	
	`str.indexOf(notAllowed) < 0`

Instead of using the `indexOf` method, students used a loop with the substring method to traverse through the string. In most cases, responses failed to structure the loop bounds correctly when traversing.
 	
`for (int i = 0; i < str.length() -`
`  notAllowed.length(); i++)`
`{`
`  if (str.substring(i, i + notAllowed.`
`  length().equals(notAllowed))`
`  {`
`  return false;`
`  }`
`}`
	`str.indexOf(notAllowed) == -1`
	
	If a loop is used to traverse the string, it must examine all substrings.
	
	`for (int i = 0; i < str.length() -`
	`  notAllowed.length() + 1; i++)`
	`{`
	`  if (str.substring(i, i + notAllowed.`
	`  length().equals(notAllowed))`
	`  {`
	`  return false;`
	`  }`
	`}`

Common Misconceptions/Knowledge Gaps of: *Write program code to satisfy methods using expressions, conditional statements, and iterative statements.*	*Responses that Demonstrate Understanding*

Responses failed to check the boundary values of `minLength` and `maxLength` when testing the length of the string parameter.
 	
`str.length() > minLength && str.length() < maxLength`
 	
Responses frequently would compare the string length to the default values of 6 and 20.
 	
`str.length() >= 6 && str.length() <= 20`
 	
or
 	
`str.length() < 6 || str.length() > 20`
	Comparison to determine if the length is within bounds:
	
	`str.length() >= minLength && str.length() <= maxLength`
	
	Comparison to determine if the length is out of bounds:
	
	`str.length() < minLength || str.length() > maxLength`

Responses only checked one condition when returning a value.

```
if (str.length() < minLength || str.length() > maxLength)
{
   return false;
}
else
{
   return true;
}
```

or

```
if (str.indexOf(notAllowed) == -1)
{
   return true;
}
else
{
   return false;
}
```

Students who used nested conditionals often did not return a value under all circumstances.

```
if (str.length() < minLength || str.length() > maxLength)
{
   return false;
}
else if (str.indexOf(notAllowed) != -1)
{
   return false;
}
```

```
if (str.length() < min || str.length() > max)
{
   return false;
}
if (str.indexOf(notAllowed) != -1)
{
   return false;
}
return true;
```

or

```
if (str.length() < minLength || str.length() > maxLength)
{
   return false;
}
else if (str.indexOf(notAllowed) != -1)
{
   return false;
}
else
{
   return true;
}
```

Based on your experience at the AP® Reading with student responses, what advice would you offer to teachers to help them improve the student performance on the exam?

Write program code to define a new type by creating a class.

- Students should pay close attention to the visibility and signatures of methods, especially when inherited or implemented.
    - Have students practice writing classes without the aid of a computer. Students can become dependent on IDEs and compilers and do not learn the proper structure and syntax for class and method headers.
    - Stress the importance of encapsulation and visibility. All instance variables should be declared as `private`.
- Students need to know how to write constructors and be sure to include the implementation of all constructors in their class.
    - Emphasize that constructors do not return values and have no return type.
    - Have students create a constructor that is complex and requires more than just assigning values to the instance variables, such as using an array or list parameter to construct and populate an instance variable.
- Students need to be aware that if a class implements an interface, then all methods in that interface need to be implemented.

Write program code to create objects of a class and call methods.

- Students should be familiar with the `String` methods in the AP Java subset, especially the `indexOf` method.

Write program code to satisfy methods using expressions, conditional statements, and iterative statements.

- Students should practice determining the proper lower bound and upper bound for loops.
    - Assign problems that have loops requiring different lower (do not start at zero) and/or different upper bounds (different than < length).
- Students should be familiar with conditionals and nested conditionals.

What resources would you recommend to teachers to better prepare their students for the content and skill(s) required on this question?

Suggested resources include:

Write program code to define a new type by creating a class.

- The Runestone Interactive Java Review offers an interactive environment for students to practice course content. Specifically, the Java Basics: Classes and Objects section and the Object-Oriented Concepts section would be most helpful for this question. This resource can be found here: http://interactivepython.org/runestone/static/JavaReview/index.html

Question #4

Task: 2D Array Processing
Topic: Latin Squares
Max. Points: 9
Mean Score: 4.75

What were the responses to this question expected to demonstrate?

This question tested the student's ability to:

- *Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` objects; and*
- *Write program code to create, traverse, and manipulate elements in 2D array objects; and*
- *Write program code to create objects of a class and call methods.*

The students were expected to write two static methods of an enclosing `ArrayTester` class. Additionally, students were required to use method `getColumn` from part (a) and two already-implemented methods of the `ArrayTester` class, `containsDuplicates` and `hasAllValues`, in their part (b) solutions.

In part (a), students were asked to construct a one-dimensional array and copy the items from a specified column of a given two-dimensional array into the new array. Students were expected to be able to construct an array with the correct number of elements, which is the number of rows in the two-dimensional array, not the number of columns. Once the array was constructed, students were expected to write a loop that accesses each item in the given column and assigns it to the corresponding element of the new array.

In part (b), students were given a square two-dimensional array and asked to evaluate if the two-dimensional array was a Latin square. A two-dimensional array of integers is a Latin square if:

- the first row has no duplicates,
- all values in the first row of the square appear in every row of the square, and
- all values in the first row of the square appear in every column of the square.

How well did the response address the course content related to this question? How well did the responses integrate the skills required on this question?

Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` objects.

> The responses demonstrated a solid grasp of constructing an integer array of the correct size to hold all items found in a specified column of a two-dimensional array as well as accessing all elements in the array. A few responses created arrays of the wrong size, using `arr2D[0].length` instead of `arr2D.length`.

Write program code to create, traverse, and manipulate elements in 2D array objects.

> Most responses correctly traversed the specified column of `arr2D` and copied over the contents of the column to the newly constructed array. Most responses that implemented the specified algorithm for determining a Latin square did so correctly, but some responses did not implement the complete algorithm and only compared rows or only compared columns.

Write program code to create objects of a class and call methods.

> Most responses called the provided methods, `containsDuplicates` and `hasAllValues`, correctly. However, some responses used the two-dimensional array `square` instead of the one-dimensional array `square[0]` or `square[r]` as the required parameter.

What common student misconceptions or gaps in knowledge were seen in the responses to this question?

Common Misconceptions/Knowledge Gaps of: *Write program code to create, traverse, and manipulate elements in 1D array or* `ArrayList` *objects.*	*Responses that Demonstrate Understanding*

Responses used the number of columns, instead of the number of rows, as the size of the array to store the values from one column of a two-dimensional array.
 	
`int[] column = new int[arr2D.length[0]];`
 	
Responses created an `ArrayList` to store the values from one column of a two-dimensional array.
 	
`ArrayList<Integer> list = new ArrayList<Integer>();`
	`int[] column = new int[arr2D.length];`

Common Misconceptions/Knowledge Gaps of: *Write program code to create, traverse, and manipulate elements in 2D array objects.*	*Responses that Demonstrate Understanding*

Responses used the `add` method of the `ArrayList` class to assign elements from the two-dimensional array to the corresponding elements in the new array.
 	
`column.add(arr2D[i][c]);`
 	
Responses switched the row and column indices when assigning elements from the two-dimensional array to the corresponding elements in the new array. In this example, the student is extracting **row** `c`, not **column** `c`.
 	
`column[i] = arr2D[c][i];`
	`column[i] = arr2D[i][c];`

Responses used nested loops when assigning elements from the two-dimensional array to the corresponding elements in the new array.
 	
`for (int r = 0; r < arr2D.length; r++)`
`{`
`  for (int j = 0; j < c; j++)`
`  {`
`  column[r] = arr2D[r][j];`
`  }`
`}`
 	
`//` correct, but inefficient
`for (int r = 0; r < arr2D.length; r++)`
`{`
`  for (int j = 0; j < arr2D.length; j++)`
`  {`
`  if (j == c)`
`  {`
`  column[r] = arr2D[r][j];`
`  }`
`  }`
`}`
	`for (int i = 0; i < arr2D.length; i++)`
	`{`
	`  column[i] = arr2D[i][c];`
	`}`

Responses copied a row of the two-dimensional array and used the copy throughout the solution.
 	
`int[] row = new int[square.length];`
 	
`for (int r = 0; r < square.length; r++)`
`{`
`  row[r] = arr2D[r][0];`
`}`
	`int[] row = square[0];`
	
	or
	
	`int[] row = new int[square.length];`
	
	`for (int c = 0; c < square.length; c++)`
	`{`
	`  row[c] = arr2D[0][c];`
	`}`

Common Misconceptions/Knowledge Gaps of: *Write program code to create objects of a class and call methods.*	*Responses that Demonstrate Understanding*

Responses called static methods using inappropriate objects.
 	
`if (this.containsDuplicates(square[0]))`
`if (square.containsDuplicates(square[0]))`
	`if (containsDuplicates(square[0]))`

Responses did not return the result of checking all rows (or columns) in a loop.
 	
`for (int r = 0; r < square.length; r++)`
`{`
`  if (hasAllValues(square[0], square[r]))`
`  {`
`  return true;`
`  }`
`  else`
`  {`
`  return false;`
`  }`
`}`
 	
or
 	
`for (int r = 0; r < square.length; r++)`
`{`
`  if (hasAllValues(square[0], square[r]))`
`  {`
`  return true;`
`  }`
`}`
`//` after all checks are made
`return false;`
 	
or
	`for (int r = 0; r < square.length; r++)`
	`{`
	`  if (!hasAllValues(square[0],square[r]))`
	`  {`
	`  return false;`
	`  }`
	`}`
	`//` after all Latin square conditions are met
	`return true;`

```
boolean flag = false;
for (int r = 0; r < square.length; r++)
{
   if (hasAllValues(square[0], square[r]))
   {
      flag = true;
   }
   else
   {
      flag = false;
   }
}
// after all checks are made
return flag;
```

Based on your experience at the AP® Reading with student responses, what advice would you offer to teachers to help them improve the student performance on the exam?

Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` *objects; and*
Write program code to create, traverse, and manipulate elements in 2D array objects.

- Students need to understand that a two-dimensional array is an array of one-dimensional arrays.
    - Develop solutions that access one row of the two-dimensional array, i.e., `square[i]`
    - Develop solutions that copy one column of the two-dimensional array
    - Develop solutions that traverse a two-dimensional array in row major order
    - Develop solutions that traverse a two-dimensional array in column major order
- Students should understand that some algorithms have two stopping conditions. One condition can be detected within the loop, but the other condition can only be detected after all checks have been made and the loop has completed.
    - Create problems that require students to implement loops with multiple terminating conditions. Sequential search is one such example.
    - Provide students with a prewritten loop containing conditionals that terminate the loop by invoking the break statement. Have the students rewrite the code to produce the same result without invoking break.

Write program code to create objects of a class and call methods.

- Students need practice using methods provided to them instead of re implementing them.
    - Develop solutions to call methods when given only the method signature
    - Develop solutions implementing and invoking static methods

What resources would you recommend to teachers to better prepare their students for the content and skill(s) required on this question?

Suggested resources include:

Write program code to create, traverse, and manipulate elements in 2D array objects.

- 2015 AP Computer Science FRQ, Question 1 requires students to write a static method to sum the contents of a one-dimensional array in part (a) and then use this method to sum all of the rows of a two-dimensional array in part (b): https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap15_frq_computer_science_a.pdf
- The current AP Computer Science Course Description, Appendix A, Page 65, Note #8 discusses what teachers and students must know about the implementation of two-dimensional arrays in the AP Computer Science A Java subset: https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-description.pdf?course=ap-computer-science-a
- The current AP Computer Science Course Description contains sample multiple choice questions and free response questions. Questions 19 and 24 test two-dimensional array concepts. Free Response Question 4 involves manipulation of one-dimensional and two-dimensional arrays. https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-description.pdf?course=ap-computer-science-a