

AP[®] COMPUTER SCIENCE A

2018 SCORING GUIDELINES

Question 2: Word Pair

Part (a)	WordPairList	5 points
----------	--------------	----------

Intent: Form pairs of strings from an array and add to an `ArrayList`

- +1 Creates new `ArrayList` and assigns to `allPairs`
- +1 Accesses all elements of `words` (*no bounds errors*)
- +1 Constructs new `WordPair` using distinct elements of `words`
- +1 Adds all necessary pairs of elements from word array to `allPairs`
- +1 **On exit:** `allPairs` contains all necessary pairs and no unnecessary pairs

Part (b)	numMatches	4 points
----------	------------	----------

Intent: Count the number of pairs in an `ArrayList` that have the same value

- +1 Accesses all elements in `allPairs` (*no bounds errors*)
- +1 Calls `getFirst` or `getSecond` on an element from list of pairs
- +1 Compares first and second components of a pair in the list
- +1 Counts number of matches of pair-like values

Question-Specific Penalties

- 1 (z) Constructor returns a value

AP[®] COMPUTER SCIENCE A

2018 SCORING GUIDELINES

Question 2: Scoring Notes

Part (a) WordPairList		5 points	
Points	Rubric Criteria	Responses earn the point if they...	Responses will not earn the point if they...
+1	Creates new <code>ArrayList</code> and assigns to <code>allPairs</code>	<code>allPairs = new ArrayList();</code> <code>allPairs = new ArrayList<>();</code> <code>this.allPairs = ...</code>	initialize a local variable that is never assigned to <code>allPairs</code>
+1	Accesses all elements of <code>words</code> (<i>no bounds errors</i>)		
+1	Constructs new <code>WordPair</code> using distinct elements of <code>words</code>		
+1	Adds all necessary pairs of elements from word array to <code>allPairs</code>	- have a loop bounds error - add unnecessary pairs	- improperly add to an <code>ArrayList</code>, e.g., <code>allPairs.get(i) = x;</code> - only add consecutive pairs (<code>words[i], words[i+1]</code>)
+1	On exit: <code>allPairs</code> contains all necessary pairs and no unnecessary pairs	- improperly add to an <code>ArrayList</code>, e.g., <code>allPairs.get(i) = x;</code> - have a loop bounds error	add pairs (i, i) or (i, j) where $i > j$
Part (b) numMatches		4 points	
Points	Rubric Criteria	Responses earn the point if they...	Responses will not earn the point if they...
+1	Accesses all elements in <code>allPairs</code> (<i>no bounds errors</i>)		access elements of <code>allPairs</code> as array elements (e.g., <code>allPairs[i]</code>)
+1	Calls <code>getFirst</code> or <code>getSecond</code> on an element from list of pairs		
+1	Compares first and second components of a pair in the list		compare using <code>==</code>
+1	Counts number of matches of pair-like values		fail to initialize the counter

Return is not assessed in part (b).

**AP[®] COMPUTER SCIENCE A
2018 SCORING GUIDELINES****Question 2: Word Pair***Part (a)*

```
public WordPairList(String[] words)
{
    allPairs = new ArrayList<WordPair>();

    for (int i = 0; i < words.length-1; i++)
    {
        for (int j = i+1; j < words.length; j++)
        {
            allPairs.add(new WordPair(words[i], words[j]));
        }
    }
}
```

Part (b)

```
public int numMatches()
{
    int count = 0;

    for (WordPair pair: allPairs)
    {
        if (pair.getFirst().equals(pair.getSecond()))
        {
            count++;
        }
    }
    return count;
}
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.