

**Question 1: Methods and Control Structures****9 points****Canonical solution****(a)**

```
public int scoreGuess(String guess)
{
    int count = 0;

    for (int i = 0; i <= secret.length() - guess.length(); i++)
    {
        if (secret.substring(i, i + guess.length()).equals(guess))
        {
            count++;
        }
    }

    return count * guess.length() * guess.length();
}
```

**5 points****(b)**

```
public String findBetterGuess(String guess1, String guess2)
{
    if (scoreGuess(guess1) > scoreGuess(guess2))
    {
        return guess1;
    }
    if (scoreGuess(guess2) > scoreGuess(guess1))
    {
        return guess2;
    }
    if (guess1.compareTo(guess2) > 0)
    {
        return guess1;
    }
    return guess2;
}
```

**4 points**

(a) scoreGuess

| Scoring Criteria   |                                                                                                                                                                                             | Decision Rules                                                                                                                                                                                                                                                                                                                                                                                                       |          |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 1                  | Compares <code>guess</code> to a substring of <code>secret</code>                                                                                                                           | Responses <b>can</b> still earn the point even if they only call <code>secret.indexOf(guess)</code><br><br>Responses <b>will not</b> earn the point if they use <code>==</code> instead of <code>equals</code>                                                                                                                                                                                                       | 1 point  |
| 2                  | Uses a substring of <code>secret</code> with correct length for comparison with <code>guess</code>                                                                                          | Responses <b>can</b> still earn the point even if they <ul style="list-style-type: none"> <li>only call <code>secret.indexOf(guess)</code></li> <li>use <code>==</code> instead of <code>equals</code></li> </ul>                                                                                                                                                                                                    | 1 point  |
| 3                  | Loops through all necessary substrings of <code>secret</code> ( <i>no bounds errors</i> )                                                                                                   | Responses <b>will not</b> earn the point if they skip overlapping occurrences                                                                                                                                                                                                                                                                                                                                        | 1 point  |
| 4                  | Counts number of identified occurrences of <code>guess</code> within <code>secret</code> ( <i>in the context of a condition involving both <code>secret</code> and <code>guess</code></i> ) | Responses <b>can</b> still earn the point even if they <ul style="list-style-type: none"> <li>initialize count incorrectly or not at all</li> <li>identify occurrences incorrectly</li> </ul>                                                                                                                                                                                                                        | 1 point  |
| 5                  | Calculates and returns correct final score ( <i>algorithm</i> )                                                                                                                             | Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>initialize count incorrectly or not at all</li> <li>fail to use a loop</li> <li>fail to compare <code>guess</code> to multiple substrings of <code>secret</code></li> <li>count the same matching substring more than once</li> <li>use a changed or incorrect <code>guess</code> length when computing the score</li> </ul> | 1 point  |
| Total for part (a) |                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                      | 5 points |

(b) `findBetterGuess`

| Scoring Criteria            |                                                                                             | Decision Rules                                                                                                                                                                                                                                                                                                                                                                                 |          |
|-----------------------------|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| 6                           | Calls <code>scoreGuess</code> to get scores for <code>guess1</code> and <code>guess2</code> | Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>fail to include parameters in the method calls</li> <li>call the method on an object or class other than <code>this</code></li> </ul>                                                                                                                                                                  | 1 point  |
| 7                           | Compares the scores                                                                         | Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>only compare using <code>==</code> or <code>!=</code></li> <li>fail to use the result of the comparison in a conditional statement</li> </ul>                                                                                                                                                          | 1 point  |
| 8                           | Determines which of <code>guess1</code> and <code>guess2</code> is alphabetically greater   | Responses <b>can</b> still earn the point even if they reverse the comparison<br><br>Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>reimplement <code>compareTo</code> incorrectly</li> <li>use result of <code>compareTo</code> as if <code>boolean</code></li> </ul>                                                                                | 1 point  |
| 9                           | Returns the identified <code>guess1</code> or <code>guess2</code> ( <i>algorithm</i> )      | Responses <b>can</b> still earn the point even if they <ul style="list-style-type: none"> <li>call <code>scoreGuess</code> incorrectly</li> <li>compare strings incorrectly</li> </ul><br>Responses <b>will not</b> earn the point if they <ul style="list-style-type: none"> <li>reverse a comparison</li> <li>omit either comparison</li> <li>fail to return a guess in some case</li> </ul> | 1 point  |
| Total for part (b)          |                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                | 4 points |
| Question-specific penalties |                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                |          |
| None                        |                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                |          |
| Total for question 1        |                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                | 9 points |

## 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 ( $\times$  •  $\div$   $\leq$   $\geq$   $<>$   $\neq$ )
- `[]` 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.*