



Eighth Edition

Be Prepared
for the

AP

Computer Science
Exam in Java

Chapter 6: Annotated Solutions
to Past Free-Response Questions

2018

Maria Litvin

Phillips Academy, Andover, Massachusetts

Gary Litvin

Skylight Publishing, Andover, Massachusetts

Skylight Publishing
Andover, Massachusetts

**Copyright © 2022 by
Maria Litvin, Gary Litvin, and Skylight Publishing**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the authors and Skylight Publishing.

Library of Congress Control Number: 2021950662

ISBN 978-0-9972528-7-3

Skylight Publishing
9 Bartlet Street, Suite 70
Andover, MA 01810

web: www.skylit.com
e-mail: sales@skylit.com
support@skylit.com

The free-response questions for this exam are posted on apstudent.collegeboard.org and, for teachers, on AP Central:

- For students: apstudent.collegeboard.org
- For teachers: apcentral.collegeboard.org/courses

Scoring guidelines are usually posted over the summer.

The www.skylit.com/beprepared/x2018all.zip file contains complete Java classes that include solutions and test programs for runnable projects.

Question 1

Part (a)

```
public boolean simulate()
{
    int pos = 0;

    for (int hops = 0; hops < maxHops; hops++) 1
    {
        pos += hopDistance();

        if (pos >= goalDistance)
            return true;
        else if (pos < 0)
            return false;
    }

    return false;
}
```

Notes:

1. Or, if you prefer,

```
for (int hops = 1; hops <= maxHops; hops++)
```

Part (b)

```
public double runSimulations(int num)
{
    int count = 0; 1

    for (int n = 0; n < num; n++)
        if (simulate())
            count++;

    return (double)count / num;
}
```

Notes:

1. Some people prefer `double count = 0;` to avoid the cast in the final ratio.

Question 2

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())) 1
            count++;

    return count;
}
```

Notes:

1. The `WordPair` class in the question does not show or mention any fields. Fields are required to hold the first and the second words of the pair, and these fields must be private. For example,

```
private String first, second;
```

Or:

```
private String[] words;
```

Regardless of how the fields are defined in `WordPair`, you cannot refer to them directly in your solution in `WordPairList` code because the fields are private in `WordPair`. You must call the `getFirst` and `getLast` accessor methods..

Question 3

```
public class CodeWordChecker implements StringChecker
{
    private int minLength, maxLength;
    private String excluded;

    public CodeWordChecker(int min, int max, String str)
    {
        minLength = min;
        maxLength = max;
        excluded = str;
    }

    public CodeWordChecker(String str)
    {
        minLength = 6;
        maxLength = 20;
        excluded = str;
    }1

    public boolean isValid(String str)
    {
        return str.length() >= minLength && str.length() <=
            maxLength && str.indexOf(excluded) == -1;
    }2
}
```

Notes:

1. You can write simply

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

and receive full credit, even though calling another constructor of the same class using this is not in the AP subset.

2. Or:

```
public boolean isValid(String str)
{
    if (str.length() < minLength || str.length() >
        maxLength || str.indexOf(excluded) >= 0)
        return false;

    return true;
}
```

Question 4**Part (a)**

```
public static int[] getColumn(int[][] arr2D, int c)
{
    int[] column = new int[arr2D.length];

    for (int r = 0; r < column.length; r++)
        column[r] = arr2D[r][c];

    return column;
}
```

Part (b)

```
public static boolean isLatin(int[][] square)
{
    if (containsDuplicates(square[0]))
        return false;

    for (int r = 1; r < square.length; r++)
        if (!hasAllValues(square[0], square[r]))
            return false;

    for (int c = 0; c < square[0].length; c++)1
        if (!hasAllValues(square[0], getColumn(square, c)))
            return false;

    return true;
}2
```

Notes:

1. Or

```
for (int c = 0; c < square.length; c++)
```

since square is a square array.

2. Or a little shorter, combining the two loops into one (as proposed by Doug Vermes):

```
public static boolean isLatin(int[][] square)
{
    if (containsDuplicates(square[0]))
        return false;
    for (int i = 0; i < square.length; i++)
        if (!hasAllValues(square[0], square[i]) ||
            !hasAllValues(square[0], getColumn(square, i)))
            return false;

    return true;
}
```