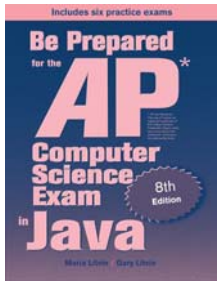


*Eighth Edition*

# Be Prepared
for the
# *AP*
# Computer Science Exam in Java

## Chapter 6: Annotated Solutions to Past Free-Response Questions

## 2015

**Maria Litvin**
Phillips Academy, Andover, Massachusetts

**Gary Litvin**
Skylight Publishing, Andover, Massachusetts

Skylight Publishing
Andover, Massachusetts

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/x2015all.zip file contains complete Java classes that include solutions and test programs for runnable projects.

# Question 1

## Part (a)

```
public static int arraySum(int[] arr)
{
  int sum = 0;

  for (int x : arr)
    sum += x;

  return sum;
}
```

## Part (b)

```
public static int[] rowSums(int[][] arr2D)
{
  int[] sums = new int[arr2D.length];

  for (int k = 0; k < sums.length; k++)
    sums[k] = arraySum(arr2D[k]);

  return sums;
}
```

## Part (c)

```
public static boolean isDiverse(int[][] arr2D)
{
  int[] sums = rowSums(arr2D);

  for (int k = 1; k < sums.length; k++)
    for (int j = 0; j < k; j++)
      if (sums[k] == sums[j])
        return false;

  return true;
}
```

## Question 2

```
public class HiddenWord
{
  private String word;

  public HiddenWord(String w)
  {
    word = w;
  }

  public String getHint(String s)
  {
    String hint = "";

    for (int i = 0; i < s.length(); i++)
    {
      String letter = s.substring(i, i+1);

      if (letter.equals(word.substring(i, i+1))) [1]
        hint += letter;
      else if (word.indexOf(letter) >= 0)
        hint += "+";
      else
        hint += "*";
    }

    return hint;
  }
}
```

**Notes:**

1. Not `letter == word.substring(i, i+1)`

# Question 3

## Part (a)

```
public int getValueAt(int row, int col)
{
  for (SparseArrayEntry e : entries)
    if (e.getRow() == row && e.getCol() == col)
      return e.getValue();

  return 0;
}
```

## Part (b)

```
public void removeColumn(int col)
{
  int i = 0;

  while (i < entries.size())
  {
    SparseArrayEntry e = entries.get(i);
    int c = e.getCol();

    if (c == col)
      entries.remove(i);
    else
    {
      if (c > col)
        entries.set(i, new SparseArrayEntry(e.getRow(),
                                            c-1, e.getValue()));
      i++;
    }
  }
  numCols--;
}
```
[1]

**Notes:**

1. Alternative solution:

```
public void removeColumn(int col)
{
  for (int i = entries.size() - 1; i >= 0; i--)
  {
    SparseArrayEntry e = entries.get(i);
    int c = e.getCol();

    if (c == col)
      entries.remove(i);
    else if (c > col)
        entries.set(i, new SparseArrayEntry(e.getRow(),
                                            c-1, e.getValue()));
  }
  numCols--;
}
```

# Question 4

## Part (a)

```
public interface NumberGroup
{
  boolean contains(int x);
}
```

## Part (b)

```
public class Range implements NumberGroup
{
  private int minValue, maxValue;

  public Range(int min, int max)
  {
    minValue = min;
    maxValue = max;
  }

  public boolean contains (int x)
  {
    return x >= minValue && x <= maxValue;
  }
}
```

## Part (c)

```
public boolean contains(int num)
{
  for (NumberGroup g : groupList)
    if (g.contains(num))
      return true;

  return false;
}
```