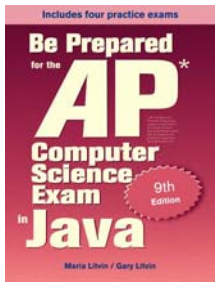

Ninth Edition

Be Prepared
for the
AP
Computer Science Exam in Java

Chapter 5: Annotated Solutions to Past Free-Response Questions

2026

Maria Litvin
Phillips Academy, Andover, Massachusetts

Gary Litvin
Skylight Publishing, Andover, Massachusetts

Skylight Publishing
Andover, Massachusetts

ISBN 978-0-9972528-3-5

Skylight Publishing
14 Lincoln Street
Andover, MA 01810

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

www.skylit.com/beprepared/x2026all.zip contains complete Java code, including solutions and test programs for runnable projects.

The free-response questions for this exam are posted on AP Central: https://apcentral.collegeboard.org/courses/ap-computer-science-a/exam/past-exam-questions

Scoring guidelines for teachers are usually posted over the summer.

Question 1

Part (a)

```
public Account(String requestedName)
{
  int num = 1;
  username = requestedName;

  while (!isAvailable(username))
  {
    username = requestedName + num; [1]
    num++;
  }
}
```

Notes:

1. Not

```
  username += num;
```

Part (b)

```
public String getShortenedName()
{
  String result = username;
  int pos = 0;

  while (pos >= 0) [1]
  {
    pos = result.indexOf("-");
    if (pos >= 0)
      result = result.substring(0, pos-1) +
                                      result.substring(pos+1);
  }
  return result;
} [2]
```

Notes:

1. According to the method's precondition, `pos` cannot be 0 — only negative or greater than 0.
2. A recursive solution:

```
// Helper method:

private String getShortenedName(String name)
{
  int pos = name.indexOf("-");

  if (pos < 0)
    return name;

  name = name.substring(0, pos-1) + name.substring(pos+1);
  return getShortenedName(name);
}

public String getShortenedName()
{
  return getShortenedName(username);
}
```

Question 2

```
public class Bottle
{
  private final double capacity;
  private double amount;

  public Bottle(double cap)
  {
    capacity = cap;
    amount = capacity;
  }

  public double updateAmount(double amtToBeRemoved) [1]
  {
    amount -= amtToBeRemoved;

    if (amount < 0.25*capacity)
      amount = capacity;

    return amount;
  }
}
```

Notes:

1. Do not recheck the given preconditions.

Question 3

```
public int moreHistoryThanMathAbsences()
{
  int numStudents = 0;

  for (CourseRecord histRec : historyList)
  {
    for (CourseRecord mathRec : mathList) [1]
    {
      if (histRec.getStudentID().equals(mathRec.getStudentID())
                && histRec.getAbsences() > mathRec.getAbsences())
      {
        numStudents++;
      }
    }
  }
  return numStudents;
}
```

Notes:

1. An `ArrayList` has the `indexOf` method. However, to use it here on `mathList`, we have to be sure that the `CourseRecord` class has the appropriate `equals` method defined. This is not guaranteed, so you might lose points if you use `mathList`'s `indexOf`. It is safer to use nested loops.

Question 4

```
public int getPointsForRow(int targetRow)
{
  boolean sameColor = true;
  String prevColor = "";[1]

  int points = 0;

  for (int col = 0; col < board[targetRow].length; col++)
  {
    Space sp = board[targetRow][col];
    points += sp.getPoints();
    String color = sp.getColor();

    if (col > 0 && !color.equals(prevColor))
      sameColor = false;

    prevColor = color;
  }

  if (sameColor)
    points *= 2;

  return points;
}[2]
```

Notes:

1. Can be initialized to `null` or any `String` value.
2. Alternative solution, with a "for-each" loop:

```
public int getPointsForRow(int targetRow)
{
  boolean sameColor = true;
  String firstColor = board[targetRow][0].getColor();
  int points = 0;

  for (Space sp : board[targetRow])
  {
    points += sp.getPoints();

    if (!sp.getColor().equals(firstColor))
      sameColor = false;
  }

if (sameColor)
  points *= 2;

return points;
}
```