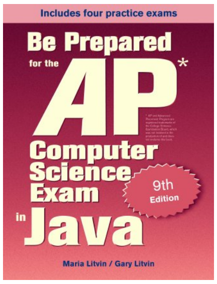

Ninth Edition

Be Prepared
for the
AP
Computer Science Exam in Java

Chapter 5: Annotated Solutions to Past Free-Response Questions

2011

Maria Litvin
Phillips Academy, Andover, Massachusetts

Gary Litvin
Skylight Publishing, Andover, Massachusetts

Skylight Publishing
Andover, Massachusetts

ISBN 978-0-9972528-3-5

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/x2011all.zip file contains complete Java classes that include solutions and test programs for runnable projects.

Question 1

Part (a)

```
public int limitAmplitude(int limit)
{
  int count = 0;

  for (int i = 0; i < samples.length; i++)
  {
    if (samples[i] > limit)
    {
      samples[i] = limit;
      count++;
    }
    else if (samples[i] < -limit)
    {
      samples[i] = -limit;
      count++;
    }
  }
  return count;
}
```

Part (b)

```
public void trimSilenceFromBeginning()
{
  int n = 0;
  while (samples[n] == 0)
    n++;

  int[] newSamples = new int[samples.length - n];

  for (int i = 0; i < newSamples.length; i++) 1
    newSamples[i] = samples[i+n];

  samples = newSamples;
```

Notes:

1. The array copying part of the code can be implemented in many different ways. For example:

```
...
for (int i = 0; i < newSamples.length; i++)
{
  newSamples[i] = samples[n];
  n++;
}
samples = newSamples;
```

or:

```
...
for (int i = 0; i < newSamples.length; i++, n++)
  newSamples[i] = samples[n];
samples = newSamples;
```

or simply:

```
...
samples = Arrays.copyOfRange(samples, n, samples.length);
```

(`Arrays.copyOfRange` is not in the AP subset, but the question doesn't say it can't be used, so this solution should receive full credit.)

Question 2

```
public class AttractiveCritter extends Critter
{
  public ArrayList<Actor> getActors()
  {
    Grid<Actor> gr = getGrid();
    ArrayList<Location> locs = gr.getOccupiedLocations();
    ArrayList<Actor> actors = new ArrayList<Actor>();

    for (Location loc : locs)
    {
      Actor a = gr.get(loc);
      if (a != this)
        actors.add(a);
    } 1

    return actors;
  }

  public void processActors(ArrayList<Actor> actors)
  {
    Grid<Actor> gr = getGrid();
    for (Actor a : actors)
    {
      Location loc = a.getLocation();
      int dir = loc.getDirectionToward(getLocation());
      Location newLoc = loc.getAdjacentLocation(dir);
      if (gr.isValid(newLoc) && gr.get(newLoc) == null)
        a.moveTo(newLoc);
    }
  }
}
```

Notes:

1. Or:

```
for (Location loc : locs)
  actors.add(gr.get(loc));
actors.remove(this);
```

While `List`'s `remove(Object obj)` method is not in the AP subset, you can use it if you do it correctly.

Question 3

Part (a)

```
public int nextTankToFill(int threshold)
{
  int iMin = filler.getCurrentIndex();

  for (int i = 0; i < tanks.size(); i++)
    if (tanks.get(i).getFuelLevel() <= threshold &&
                               tanks.get(i).getFuelLevel() <
                                    tanks.get(iMin).getFuelLevel())  1
      iMin = i;

  return iMin;
}
```

Notes:

1. Or <=.

Part (b)

```
public void moveToLocation(int locIndex)
{
  int index = filler.getCurrentIndex();
  if (locIndex > index)
  {
    if (!filler.isFacingRight())
      filler.changeDirection();
    filler.moveForward(locIndex - index);
  }
  else if (locIndex < index)
  {
    if (filler.isFacingRight())
      filler.changeDirection();
    filler.moveForward(index - locIndex);
  }
}
```

Question 4

Part (a)

```
private void fillBlock(String str)
{
  int i = 0;

  for (int r = 0; r < numRows; r++)
  {
    for (int c = 0; c < numCols; c++)
    {
      if (i < str.length())
        letterBlock[r][c] = str.substring(i, i+1);
      else
        letterBlock[r][c] = "A";
      i++;
    }
  }
}
```

Part (b)

```
public String encryptMessage(String message)
{
  String code = "";
  while (message.length() > 0)
  {
    int n = numRows * numCols;
    if (n > message.length())
      n = message.length(); 1
    fillBlock(message.substring(0, n)); 2

    code += encryptBlock();
    message = message.substring(n);
  }

  return code;
}
```

Notes:

1. Or:

   ```
   int n = Math.min(numRows * numCols, message.length());
   ```

   although `Math.min` is not in the AP subset.

2. Or simply:

   ```
   fillBlock(message);
   ```