8 One-Dimensional Arrays

8.1

```
1  import CSLib.*;
2
3  public class Simple {
4    // Simple:  read10 - reads 10 integers
5    //          print10 - prints 10 integers in reverse
6    // Author:  Maddie Kamin, Friday, April 13, 2001
7
8    int[] counts;
9
10   public Simple () {
11      counts = new int[10];
12   }
13
14   public void read10 () {
15     InputBox in = new InputBox();
16     in.setPrompt("Enter integer:");
17     counts[0]=in.readInt();
18     counts[1]=in.readInt();
19     counts[2]=in.readInt();
20     counts[3]=in.readInt();
21     counts[4]=in.readInt();
22     counts[5]=in.readInt();
23     counts[6]=in.readInt();
24     counts[7]=in.readInt();
25     counts[8]=in.readInt();
26     counts[9]=in.readInt();
27   }
28
29   public void print10 () {
30     OutputBox out = new OutputBox("Input");
31
32     out.print(counts[9] + " " + counts[8] + " "
```

```
          + counts[7] + " " + counts[6] + " "
          + counts[5] + " " + counts[4] + " "
          + counts[3] + " " + counts[2] + " "
          + counts[1] + " " + counts[0] + "\n");
   }
}
```

8.2

```
import CSLib.*;

public class Simple {
  // Simple:  read10 - reads 10 integers
  //          print10 - prints the sum of 10 integers
  // Author:  Maddie Kamin, Friday, April 13, 2001

  int[] counts;

  public Simple () {
     counts = new int[10];
  }

  public void read10 () {
    InputBox in = new InputBox();
    in.setPrompt("Enter integer:");
    counts[0]=in.readInt();
    counts[1]=in.readInt();
    counts[2]=in.readInt();
    counts[3]=in.readInt();
    counts[4]=in.readInt();
    counts[5]=in.readInt();
    counts[6]=in.readInt();
    counts[7]=in.readInt();
    counts[8]=in.readInt();
    counts[9]=in.readInt();
  }

  public void print10 () {
    OutputBox out = new OutputBox("Input");

    out.print(counts[0] + counts[1] +
          + counts[2] + counts[3] +
          + counts[4] + counts[5] +
          + counts[6] + counts[7] +
          + counts[8] + counts[9] );
   }
}
```

8.3

1.

```
import CSLib.*;

public class Simple {
  // Simple:  read10 - reads 10 integers
  //          print10 - prints 10 integers in reverse
  // Author:  Maddie Kamin, Friday, April 13, 2001

  int[] counts;

  public Simple () {
     counts = new int[10];
  }

  public void read10 () {
    InputBox in = new InputBox();
    in.setPrompt("Enter integer:");

    for (int i=0; i<10; i++)
      counts[i]=in.readInt();
  }

  public void print10 () {
    OutputBox out = new OutputBox("Input");

    for (int i=9; i>=0; i--)
      out.print(counts[i] + " ");

    out.println();
   }
}
```

2.

```
import CSLib.*;

public class Simple {
  // Simple:  read10 - reads 10 integers
  //          print10 - prints every even one of 10 integers
  // Author:  Maddie Kamin, Friday, April 13, 2001

  int[] counts;

  public Simple () {
     counts = new int[10];
  }

```

```
  public void read10 () {
    InputBox in = new InputBox();
    in.setPrompt("Enter integer:");

    for (int i=0; i<10; i++)
      counts[i]=in.readInt();
  }

  public void print10 () {
    OutputBox out = new OutputBox("Input");

    for (int i=0; i<10; i=i+2)
      out.print(counts[i] + " ");

    out.println();
   }
}
```

3.

```
import CSLib.*;

public class Simple {
  // Simple:  read10 - reads 10 integers
  //          print10 - prints 10 integers, 5 forward and 5 backwards
  // Author:  Maddie Kamin, Friday, April 13, 2001

  int[] counts;

  public Simple () {
     counts = new int[10];
  }

  public void read10 () {
    InputBox in = new InputBox();
    in.setPrompt("Enter integer:");

    for (int i=0; i<10; i++)
      counts[i]=in.readInt();
  }

  public void print10 () {
    OutputBox out = new OutputBox("Input");

    for (int i=0; i<5; i++)
      out.print(counts[i] + " ");
    for (int i=9; i>4; i--)
      out.print(counts[i] + " ");

    out.println();
```

```
31      }
32    }
```

8.4

```
1   import CSLib.*;
2
3   public class Simple {
4     // Simple:  read10 - reads up to MAX_CLASS_SIZE integers
5     //          print10 - prints up to MAX_CLASS_SIZE integers
6     // Author:  Maddie Kamin, Friday, April 13, 2001
7
8     private final int MAX_CLASS_SIZE = 1000;
9     private int[] counts;
10    private int size = 0;
11
12    public Simple () {
13       counts = new int[MAX_CLASS_SIZE];
14    }
15
16    public void read10 () {
17      InputBox in = new InputBox();
18      in.setPrompt("Enter integer, or press OK to terminate");
19
20      while (true) {
21        counts[size] = in.readInt();
22        if (in.eoi()) break;
23        size++;
24      }
25    }
26
27    public void print10 () {
28      OutputBox out = new OutputBox("Input");
29
30      for (int i=0; i<size; i++)
31        out.print(counts[i] + " ");
32
33      out.println();
34     }
35  }
```

8.5

```
int min = 100,
    max = 0,
    nzmin = 100,
    nzeros = 0;

for (int i=0; i<size; i++) {
  // nzmin is lowest non-zero grade among grades[0]..grades[i-1]
```

```
  if (nzmin > grades[i] && grades[i] > 0)
    nzmin = grades[i];
  // min is lowest grade among grades[0]..grades[i-1]
  if (min > grades[i])
    min = grades[i];
  // max is highest grade among grades[0]..grades[i-1]
  if (max < grades[i])
    max = grades[i];
  if (grades[i] == 0)
    nzeros++;
}
```

8.6

```
int min = 100,
    max = 0,
    nzmin = 100,
    nzeros = 0;
double sum = 0.0,
       sumnz = 0.0;

for (int i=0; i<size; i++) {
  // nzmin is lowest non-zero grade among grades[0]..grades[i-1]
  if (nzmin > grades[i] && grades[i] > 0)
    nzmin = grades[i];
  // min is lowest grade among grades[0]..grades[i-1]
  if (min > grades[i])
    min = grades[i];
  // max is highest grade among grades[0]..grades[i-1]
  if (max < grades[i])
    max = grades[i];
  // sumnz is the sum of non-zero grades among grades[0]..grades[i-1]
  if (grades[i] == 0)
    nzeros++;
  else
    sumnz = sumnz + grades[i];
  // sum is the sum of all grades among grades[0]..grades[i-1]
  sum = sum + grades[i];
}
avg = sum / size;
avgnz = sumnz / (size - nzeros);
```

8.7

```
int n90 = 0;

for (int i=0; i<size; i++) {
  if (grades[i] >= 90)
    n90++;
}
```

8.8

```
int i = 0;
while (i < size && grades[i] < 80 || grades[i] > 89) {
   // a grade between 80 and 89 does not occur among grades[0] .. grades[i-1]
   i++;
}
// a grade between 80 and 89 occurs in grades if and only if i < size
```

8.9

```
int min1 = 100,
    min2 = 100,
    max1 = 0,
    max2 = 0;
double sum1 = 0.0,
       sum2 = 0.0;

for (int i=0; i<size; i++) {
  // min1 is lowest non-zero grade among grades1[0]..grades1[i-1]
  if (min1 > grades1[i] && grades1[i] > 0)
    min1 = grades1[i];
  // min2 is lowest non-zero grade among grades2[0]..grades2[i-1]
  if (min2 > grades2[i] && grades2[i] > 0)
    min2 = grades2[i];
  // max1 is highest grade among grades1[0]..grades1[i-1]
  if (max1 < grades1[i])
    max1 = grades1[i];
  // max2 is highest grade among grades2[0]..grades2[i-1]
  if (max2 < grades2[i])
    max2 = grades2[i];
  // sum1 is the sum of all grades among grades1[0]..grades1[i-1]
  sum1 = sum1 + grades1[i];
  // sum2 is the sum of all grades among grades2[0]..grades2[i-1]
  sum2 = sum2 + grades2[i];
}
double avg1 = sum1 / size;
double avg2 = sum2 / size;
```

8.10

```
public void read () {
  InputBox in = new InputBox();
  size = -1;
  in.setPrompt("Enter grades for exam 1, followed by -1:");
  // grades1[0]..grades1[size] have valid grades
  do {
```

```
    size++;
    grades1[size] = in.readInt();
  } while (grades1[size] != -1);
  // grades1[0]..grades1[size-1] have valid grades; grades1[size] is -1.

  in.setPrompt("Enter the same number of grades for exam 2:");
  for (int i=0; i<size; i++)
    grades2[i] = in.readInt();
}
```

8.11

```
int improved = 0,
    declined = 0,
    same;
for (int i=0; i < size; i++)
  if (grades1[i] < grades2[i]) improved++;
  else if (grades1[i] > grades2[i]) declined++;
same = size - improved - declined;
```

8.12

```
int improved = 0,
    declined = 0,
    same = 0;
for (int i=0; i < size; i++)
  if (grades1[i] > 0 && grades2[i] > 0)
    if (grades1[i] < grades2[i]) improved++;
    else if (grades1[i] > grades2[i]) declined++;
    else same++;
```

8.13

```
int biggest = 0,
    most_improved = 0;
for (int i=0; i < size; i++)
  if ((grades2[i]-grades1[i]) > biggest) {
    biggest = grades2[i] - grades1[i];
    most_improved = i;
  }
```

8.14

```
for (int i=0; i < size; i++)
  sum = sum + grades2[i]-grades1[i];
double avg = sum / size;
```

8.15

```
import CSLib.*;

public class GradeReport {
  // Build and use a GradeDistribution
  // Author: Robert D. Klapper, November 16, 2000

  public void readScores (GradeDistribution distr) {
    InputBox in = new InputBox("Enter grade: ");
    while (true) {
      int g = in.readInt();
      if (in.eoi()) return;
      distr.insertGrade(g);
    }
  }

  public void barGraph(GradeDistribution distr) {
    // With n groups and maximum count of m, and
    // screen width w and height h, place bar for
    // i(th) group at i*w/n pixels from left, with
    // width w/n and height h/m times the count
    DrawingBox out = new DrawingBox("Grade Histogram");
    int m = distr.maxCount(),
        n = distr.numGroups(),
        w = out.getDrawableWidth() - 10,
        h = out.getDrawableHeight() - 5;
    int barwidth = w/n;
    for (int i=0; i<n; i++) {
      int horizpos = i*barwidth,
          height = distr.groupCount(i)*h/m;
      out.drawRect(horizpos+5, h-height+5, barwidth, height);
    }
  }
}
```

8.16

The first assignment statement (`A[i] = A[j]`) destroys the value that was stored in `A[]`, before it can be assigned to `A[j]`.