6 Iteration

6.1

There are 21 integers from -10 to 10 (including 0), so the loop is executed 21 times.

6.2

```
1   import CSLib.*;
2
3   public class Temperature {
4     // Print a table of corresponding F/C temperatures.
5     // Author: Mercury Reising, June 3, 2001
6
7     public void tabulate () {
8       final double
9         LOW_TEMP = -10.0,
10        HIGH_TEMP = 10.0;
11      double
12        cent,  // The Centigrade temperature.
13        fahr;  // The Fahrenheit temperature.
14
15      OutputBox out = new OutputBox();
16      out.println("\tDEGREES F\tDEGREES C");
17
18      fahr = LOW_TEMP;
19      while (fahr <= HIGH_TEMP) {
20        cent = (fahr - 32.0) * (5.0 / 9.0); // Convert F to C
21        out.println("\t" + fahr + "\t\t" + cent);
22        fahr = fahr + 1.0;   // Increment the Fahrenheit value.
23      }
```

```
  }
}
```

6.3

```
import CSLib.*;

public class Beer {
  //  Print the song "99 Bottles of Beer on the Wall."
  //  Author: Ty Wonon, September 22, 2001

  public void sing () {

    final int MAX_BOTTLES = 99;
    int numberOnWall;

    OutputBox out = new OutputBox();
    numberOnWall = MAX_BOTTLES;

    out.println(numberOnWall + " bottles of beer on the wall,");
    out.println(numberOnWall + " bottles of beer.");
    out.println("If one of those bottles should happen to fall,");
    numberOnWall = numberOnWall - 1;
    while (numberOnWall > 1) {
      out.println(numberOnWall + " bottles of beer on the wall.");
      out.println(numberOnWall + " bottles of beer on the wall,");
      out.println(numberOnWall + " bottles of beer.");
      out.println("If one of those bottles should happen to fall,");
      numberOnWall = numberOnWall - 1;
    }
    out.println("1 bottle of beer on the wall.");
    out.println("1 bottle of beer on the wall,");
    out.println("1 bottle of beer.");
    out.println("If that one bottle should happen to fall,");
    out.println("No more bottles of beer on the wall!");
  }
}
```

6.4

```
import CSLib.*;

public class Temperature {
  // Print a table of corresponding F/C temperatures.
  // Author: Mercury Reising, June 3, 2001

  public void tabulate () {
    final double
      LOW_TEMP = -10.0,
      HIGH_TEMP = 10.0;
```

```
    double
      cent,  // The Centigrade temperature.
      fahr;  // The Fahrenheit temperature.

    OutputBox out = new OutputBox();
    out.println("\tDEGREES F\tDEGREES C");

    for (fahr = LOW_TEMP; fahr <= HIGH_TEMP; fahr = fahr + 1.0) {
      cent = (fahr - 32.0) * (5.0 / 9.0); // Convert F to C
      out.println("\t" + fahr + "\t\t" + cent);
    }
  }
}
```

6.5

```
import CSLib.*;

public class DigitReverse {
  //  Reversal of digits
  //  Author: Anna Graham, September 8, 2001

  public void reverse () {

    InputBox in = new InputBox();
    in.setPrompt("Enter a positive integer: ");
    int n = in.readInt();

    OutputBox out = new OutputBox("DIGIT REVERSAL");
    out.print("The reversal of " + n + " is ");
    while (n > 0) {
      out.print(n % 10);
      n = n / 10;
    }
    out.println();
  }
}

```

6.6

```
import CSLib.*;

public class TenInABed {
  //  Print the nursery rhyme "Ten In a Bed."
  //  Author: Leah S. Gordon, December 29, 1996

  public void sing () {

    final int MAX_NUMBER_IN_BED = 10;
```

```
10        int numberInBed;
11
12        OutputBox out = new OutputBox();
13        numberInBed = MAX_NUMBER_IN_BED;
14
15        while (true) {
16          out.println(numberInBed + " in a bed and the little one said,");
17          if (numberInBed == 1) break;
18          out.println("     \"Roll over, roll over.\"");
19          out.println("They all rolled over and one fell out,");
20          numberInBed = numberInBed - 1;
21        }
22        out.println("     \"Alone at last.\"");
23      }
24    }
```

6.7

If we rewrite the `if` statement in the `Scores` program as

```
if (maxOfScores < score)          // new largest score
   maxOfScores = score;
else if (minOfScores > score)     // new smallest score
   minOfScores = score;
```

The code may not work properly. There are a number of things that could go wrong.

1. Suppose that there are three scores entered as $0, 100, 50$. The first one entered (0) will be recorded as the maximum (so far), but the test for a new minimum would be skipped. Next, 100 is found to be a new maximum, and again the test fror a new minimum would be skipped. Finally, 50 is found to be less than `maxOfScores`, so the minimum test is performed (and satisfied), and `minOfScores` is set to 50 – the wrong value.

2. Suppose that there are any number of scores entered in strictly increasing order – such as $1, 2, 3, 5, 80, 96, 100$. Each one will satisfy the `maxOfScores<scores` test and cause `maxOfScores` to be reset. But in each case, the test for a new minimum will be skipped. Therefore, `minOfScores` will remain 999.

6.8

Computing the median of a set of scores requires that the set be sorted. This involves programming techniques that we will not study until Chapter 8.

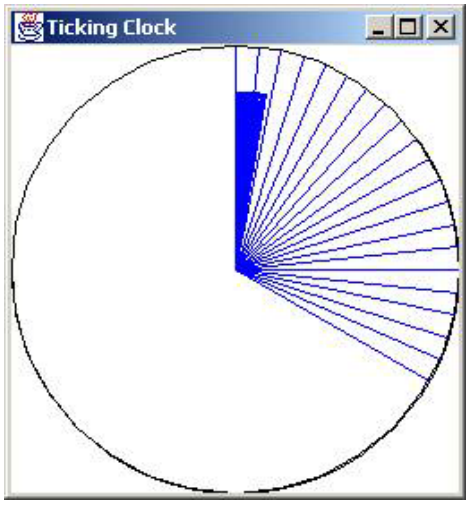
6.9

```
1   import CSLib.*;
2   import java.awt.*;
3
4   public class Surprise {
5     // Surprise with an exploding balloon
```

```
6     // Author: Rachel McDermott, September 27, 1996
7
8     DrawingBox board;
9     int width, height,
10        dmax,     // The maximum diameter circle
11        diameter; // The balloon diameter
12
13    public void setup () {
14      board = new DrawingBox("Surpriser");
15
16      width = board.getDrawableWidth();
17      height = board.getDrawableHeight();
18      dmax = (int)Math.min(width, height);
19    }
20
21    public void draw (){
22      board.setColor(Color.yellow);
23      for (diameter=0; diameter < dmax; diameter++) {
24        board.fillOval((width-diameter)/2, (height-diameter)/2,
25                       diameter, diameter);
26        Timer.pause(10);               // sleep for 10 msec
27      }
28
29      board.setColor(Color.red);
30      board.drawString("Surprise!", (width-53)/2, (height+14)/2);
31
32      Timer.pause(1000); // sleep for 1 sec
33
34      board.setColor(board.getBackground());
35      for (; diameter >= 0 ; diameter--) {
36        board.drawOval((width-diameter)/2, (height-diameter)/2,
37                       diameter, diameter);
38        Timer.pause(10);               // sleep for 10 msec
39      }
40    }
41  }
```

6.10

If you remove the `clear` call in the `Sweep` class, then the sweeping hands "paint" the clock face as they sweep. A sweep of 20 minutes (from 12:00) looks like this;



6.11

1. If we advance the clock by 65 minutes (starting at 12 : 00), then the clock records a time of 0 : 65.

2. Sweeping increases the value of the `Clock`'s `minute` instance variable. Therefore, there is danger of overflow if this continues indefinitely.

3. If we were to advance the value of the `Clock`'s `minute` instance variable by 60 every second, then it would exceed the maximum value of $2,147,483,647$ (see Section 3.6.1) in about 414 days.