SOURCE CODE EXAMPLES: JAVA

This appendix contains the code examples referred to in the main text. There is a version of the appendix for each language. This is the Java version.

#1 EXAMPLE FOR THE CALLING DIAGRAM

```
int main() {
  function_one();
  function_two();
}

int function_one() {
  function_three();
}

int function_two() {
  function_four();
  function_one();
  function_two();
}

int function_three() {}

int function_four() {
  function_three();
}
```

#2 SMALL EXAMPLE FOR HAND SIMLUATION

```
 byte v1 = 10;
 int v2 = v1 * 100;

 while (v2 > 500) {
   System.out.print(“Excess over 500 = ”);
   System.out.println(v2 - 500);
   v2 = --v1 * 100;
 }
```

#3 EXAMPLE FOR SCOPE REGIONS

```
void A() {
  int B;
}

void E() {
  int C;
   {
      int D;
   }
}
```

#4 LARGE EXAMPLE FOR HAND SIMULATION

```
class HS {
  int g1 = 100;
  int g2 = 20;

  void f1(int xx) {
    int w = 10;
    w = w + g1;
    System.out.println(w / xx);
  }

  int f2() {
    int w = 5;
    g1 = g2 + w;
    return w + 2;
  }

  void run() {
    int r;
    r = f2();
    f1(r);
    System.out.print("g1 = ");
    System.out.println(g1);
  }

  public static void main(String[] args) {
    (new Example4()).run();
  }
}
```

#5 TOP-DOWN DESIGN EXAMPLE

```
/**********************************
 The paint estimator
**********************************/
import java.io.*;

class Estimator {
  static final float paintCost = 12.5f;
  static final float height = 8.0f;

  float costOfPainting, length, width,
    coverage, area;

  void inputValues() throws java.io.IOException {
    BufferedReader instr = new BufferedReader(
           new InputStreamReader(System.in));
    System.out.println("Type in length: ");
    length = Float.valueOf(instr.readLine()).floatValue();
    System.out.println("Type in width: ");
    width = Float.valueOf(instr.readLine()).floatValue();
    System.out.println("Type in coverage: ");
    coverage = Float.valueOf(instr.readLine()).floatValue();
  }

  void calculateArea() {
    area = 2 * (length + width) * height + length * width;
  }

  void calculateCostFromArea() {
    costOfPainting = area / coverage * paintCost;
  }

  void calculateCost() {
    calculateArea();
    calculateCostFromArea();
  }
  void displayCost() {
    System.out.println("Cost of painting a room is $" +
          costOfPainting);
  }

  void run() throws java.io.IOException {
    inputValues();
    calculateCost();
    displayCost();
  }

  public static void main(String[] args) throws java.io.IOException {
    (new Estimator()).run();
  }
}
```

#6 TOP-DOWN DESIGN EXAMPLE, REWORKED

```
/**********************************
 The paint estimator
**********************************/

import java.io.*;

class Estimator {
  static final float paintCost = 12.5f;
  static final float height = 8.0f;

  float length, width, coverage;

  void inputValues() throws java.io.IOException {
    BufferedReader instr = new BufferedReader(
           new InputStreamReader(System.in));
    System.out.println("Type in length: ");
    length = Float.valueOf(instr.readLine()).floatValue();
    System.out.println("Type in width: ");
    width = Float.valueOf(instr.readLine()).floatValue();
    System.out.println("Type in coverage: ");
    coverage = Float.valueOf(instr.readLine()).floatValue();
  }
  float calculateArea(float length, float width) {
    return 2 * (length + width) * height + length * width;
  }

  float calculateCostFromArea(float area, float coverage) {
    return area / coverage * paintCost;
  }

  float calculateCost(float l, float w, float c) {
    float area;

    area = calculateArea(l, w);
    return calculateCostFromArea(area, c);
  }
  void displayCost(float l, float w, float c, float cp) {
     System.out.println("Cost of painting a room " + l + " by " + w);
     System.out.println(" with paint of coverage " + c + " is $" + cp);
  }

   void run() throws java.io.IOException {
     float costOfPainting;
     inputValues();
     costOfPainting = calculateCost(length, width, coverage);
     displayCost(length, width, coverage, costOfPainting);
  }
  public static void main(String[] args) throws java.io.IOException {
    (new Estimator()).run();
  }
}
```

#7 THE MAIN PROGRAM FOR THE TOP-DOWN DESIGN EXAMPLE

```
void run() {
   float costOfPainting, length, width, coverage;

   length = inputLength();
   width = inputWidth();
   coverage = inputCOverage();
   calculateCost(length, width, coverage);
   displayCost(length, width, coverage, costOfPainting);
}
```

#8 THE PAINT ESTIMATOR DESIGN IN CLASSES

```
class Room {
  private float Width, Length;      // in ft.
  private static final float Height = 8.0;      // in ft.
}

class Paint {
  private static final float CostOfPaint = 12.5;// in $/gallon
  float Coverage;       // in sq.ft./gallon
}

class Estimator {
  private Room room;
  private Paint paint;
  private float CostOfPainting;     // in $
}
```

#9 THE PAINT ESTIMATOR DESIGN WITH CLASS METHODS

```
class Room {
  private float Width, Length;      // in ft.
  private static final float Height = 8.0f;     // in ft.
  public Room() {} // the default constructor
  public float getWidth() { return Width; }     // accessor method
  public float getLength() { return Length; }   // accessor method
  public float getHeight() { return Height; }   // accessor method
  public void input() throws java.io.IOException {
    // input method prompts
    BufferedReader instr = new BufferedReader(
           new InputStreamReader(System.in));
    System.out.println("Type width ");
    Width = Float.valueOf(instr.readLine()).floatValue();
    System.out.println("Type length: ");
    Length = Float.valueOf(instr.readLine()).floatValue();
  }
}

// (continued)
```

```
class Paint {
  private static final float CostOfPaint = 12.5f; // in $/gallon
  private float Coverage;              // in sq.ft./gallon
  public Paint()  {}   // default constructor
  public float getCoverage() { return Coverage; } // accessor method
  public float getCost() { return CostOfPaint; } // accessor method
  public void input() throws java.io.IOException {          // input method
    BufferedReader instr = new BufferedReader(
           new InputStreamReader(System.in));
    System.out.println("Type coverage: ");
    Coverage = Float.valueOf(instr.readLine()).floatValue();
  }
}

class Estimator {
  private Room room;
  private Paint paint;
  private float CostOfPainting;     // in $
  public Estimator() {} // default constructor
  void estimate() {
    ...
  }
  void display() {
    System.out.println("Cost is " << CostOfPainting);
  }
  void run() {
    ...
  }
  public static void main(String[] args) {
    ...
  }
}
```

#10 THE PAINT ESTIMATOR TOP LEVEL METHODS

```
 public void estimate() {
    float wallArea =
      2 * (room.getWidth() + room.getLength()) * room.getHeight();
    float ceilingArea = room.getWidth() * room.getLength();
    CostOfPainting = (wallArea + ceilingArea) / paint.getCoverage() *
  paint.getCost();
  }

  public void run() throws java.io.IOException {
    room.input();
    paint.input();
    estimate();
    display();
  }

  public static void main(String[] args) throws java.io.IOException {
    (new Estimator()).run();
  }
```

#11 SIMPLE EXAMPLE FOR DEBUGGING

```
DataInputStream instr(System.in);
while (num != 9999) {
  total += num;
  num = instr.readInt();
}
```

#12 DEBUGGING WITH TRACE ADDED

```
BufferedReader instr = new BufferedReader(
                          new InputStreamReader(System.in));
while (num != 9999) {
  System.out.println("Trace total: " + total);
  total += num;
  num = Integer.parseInt(instr.readLine());
}
```

#13 ADDING AN ASSERTION

Java has no macro facility, and hence no inbuilt assertion mechanism.

#14 AN ASSERTION IN STRAIGHT CODE

```
if (x < 0 || x > 9) {
  System.out.println("Error: x is out of range " + x);
  System.exit(1);
}
```

#15 A DEBUGGING EXAMPLE

```
import java.io.*;

class Debugging {
  public static void main(String[] args) throws java.io.IOException {
    int c;
    char ch, lastch = '\0';
    int totalIEs = 0, totalEIs = 0;

    BufferedReader instr = new BufferedReader(
         new InputStreamReader(System.in));

    System.out.println("Type any number of lines followed by EOF");

    while ((c = instr.read()) != -1) {
      ch = Character.toLowerCase((char)c);
      if (ch == 'i' && lastch == 'e')
        ++totalEIs;
      else if (ch == 'e' && lastch == 'i')
        ++totalIEs;
      else
        lastch = ch;
    }
// continued
```

```
    System.out.println();
    System.out.println("the number of IEs is " + totalIEs);
    System.out.println();
    System.out.println("and the number of EIs = " + totalEIs);
  }
}
```

#16 THE EXAMPLE WITH TRACE ADDED

```
import java.io.*;

class Debugging {
  public static void main(String[] args) throws java.io.IOException {
    int c;
    char ch, lastch = '\0';
    int totalIEs = 0, totalEIs = 0;

    BufferedReader instr = new BufferedReader(
         new InputStreamReader(System.in));

    System.out.println("Type any number of lines followed by EOF");

    while ((c = instr.read()) != -1) {
      ch = Character.toLowerCase((char)c);
      if (ch == 'i' && lastch == 'e')
        ++totalEIs;
      else if (ch == 'e' && lastch == 'i')
        ++totalIEs;
      else
        lastch = ch;
      System.out.println("ch = " + ch + " lastch = " + lastch +
       " totalIEs = " + totalIEs +
       " totalEIS = " +  totalEIs);
    }
    System.out.println();
    System.out.println("the number of IEs is " + totalIEs);
    System.out.println();
    System.out.println("and the number of EIs = " + totalEIs);
  }
}
```

#17 THE EXAMPLE WITH ASSERTION ADDED

```
else
      lastch = ch;
  /*****************************************/
    if (lastch != ch) {
      System.out.println(“Error: lastch ! = ch”);
      System.exit(1);
    }
  /*****************************************/
```

```
  }
  ...
```

#18 THE EXAMPLE ASSERTION AND TRACE

```
...
    else
      lastch = ch;
  /*//////   a trace statement   ///////*/
    System.out.println("lastch = " + lastch + " ch = " + ch);
  /*////////////////////////////////////*/
  /*******   an assertion  *************/
  /*  if (lastch != ch) {
      System.out.println("Error: lastch ! = ch");
      System.exit(1);
    } */
  /***********************************/
  }
...
```

#19 THE EXAMPLE DEBUGGED

```
import java.io.*;

class Debugging {
  public static void main(String[] args) throws java.io.IOException {
    int c;
    char ch, lastch = '\0';
    int totalIEs = 0, totalEIs = 0;

    BufferedReader instr = new BufferedReader(
         new InputStreamReader(System.in));

    System.out.println("Type any number of lines followed by EOF");

    while ((c = instr.read()) != -1) {
      ch = Character.toLowerCase((char)c);
      if (ch == 'i' && lastch == 'e')
        ++totalEIs;
      else if (ch == 'e' && lastch == 'i')
        ++totalIEs;
      //else
        lastch = ch;
    }
    System.out.println();
    System.out.println("the number of IEs is " + totalIEs);
    System.out.println();
    System.out.println("and the number of EIs = " + totalEIs);
  }
}
```