Key to CSE142 Sample Final Exam, Summer 2018

1. The program produces the following output:

```
4 16 23
2 16 123
4 16 123
3 42 223
5 42 223
```

2.

Original List	Final List
[]	[]
[7]	[7]
[3, 2]	[0, 2]
[5, 4, 3]	[0, 4, 6]
[2, 4, 6, 8]	[0, 4, 12, 24]

3. The program produces the following output:

```
cass 1
cass 2
cass

denny 1
john 2
denny john

cass 1
john 2
john

michelle 1
john 2
john
```

4. One possible solution appears below.

```
public static int reportLongLines(Scanner input, int maxLength) {
    int lineCount = 0;
    int longLineCount = 0;
    while (input.hasNextLine()) {
        lineCount++;
        String line = input.nextLine();
        if (line.length() > maxLength) {
            longLineCount++;
            System.out.println("Line " + lineCount + " excess text: '" +
                               line.substring(maxLength) + "'");
        }
    }
    return longLineCount;
}
```

5. One possible solution appears below.

```
public static void formatLib(Scanner console, Scanner input,
                             PrintStream output, int maxLength) {
   int len = 0;
   while (input.hasNext()) {
      String tok = input.next();
      if (tok.startsWith("<") && tok.endsWith(">")) {
         tok = tok.substring(1, tok.length() - 1);
         System.out.print(tok + " needed: ");
         tok = console.next();
      }

      tok += " ";
      if (len + tok.length() > maxLength) {
         output.println();
         len = 0;
      }
      output.print(tok);
      len += tok.length();
   }
}
```

6. One possible solution appears below.

```
public static boolean isConsecutive(int[] a) {
    for(int i = 0; i < a.length - 1; i++) {
        if (a[i] != a[i + 1] - 1]) {
            return false;
        }
    }
    return true;
}
```

7. Two possible solutions appear below.

```
public boolean isWorkTime() {
    if (amPm.equals("AM")) {
        return hour >= 9 && hour <= 11;
    } else {  // PM
        return (1 <= hour && hour <= 4) || hour == 12 ||
                (hour == 5 && minute == 0);
    }
}

public boolean isWorkTime() {
    int totalMins = hour % 12 * 60 + minute;
    if (amPm.equals("PM")) {
        totalMins += 12 * 60;
    }
    return 9 * 60 <= totalMins && totalMins <= 17 * 60;
}
```

8. Two possible solutions appear below.

```
public class Minnow extends Critter {
    private int cycleLength;
    private int cycleStep;

    public Minnow() {
        cycleLength = 1;
        cycleStep = 0;
    }

    public boolean eat() {
        cycleLength++;
        cycleStep = 0;

        return false;
    }

    public Direction getMove() {
        if(cycleStep == 0) {
            cycleStep++;
            return Direction.SOUTH;
        } else if(cycleStep < cycleLength) {
            cycleStep++;
        } else {
            cycleStep = 0;
        }

        if(cycleLength % 2 == 1) {
            return Direction.EAST;
        } else {
            return Direction.WEST;
        }
    }
}
```

```
public class Minnow extends Critter {
    private Direction currHoriz;
    private int cycleLength;
    private int cycleStep;

    public Minnow() {
        currHoriz = Direction.EAST;
        cycleLength = 1;
        cycleStep = 0;
    }

    public boolean eat() {
        cycleLength++;
        cycleStep = 0;

        if(currHoriz == Direction.EAST) {
            currHoriz = Direction.WEST;
        } else {
            currHoriz = Direction.EAST;
        }
        return false;
    }

    public Direction getMove() {
        if(cycleStep == 0) {
            cycleStep++;
            return Direction.SOUTH;
        } else if(cycleStep < cycleLength) {
            cycleStep++;
            return currHoriz;
        } else {
            cycleStep = 0;
            return currHoriz;
        }
    }
}
```

9. Four possible solutions appear below.

```
public static int[] delta(int[] a) {
    if (a.length == 0) {
        return new int[0];
    }

    int[] result = new int[a.length + (a.length - 1)];
    for (int i = 0; i < result.length; i++) {
        if (i % 2 == 0) {
            result[i] = a[i / 2];
        } else {
            result[i] = a[i / 2 + 1] - a[i / 2];
        }
    }
    return result;
}
```

```
public static int[] delta(int[] a) {
    if (a.length == 0) {
        return new int[0];
    }

    int[] b = new int[2 * a.length - 1];
    for (int i = 0; i < a.length; i++) {
        b[2 * i] = a[i];
    }
    for (int i = 1; i < b.length; i += 2) {   // <- NOTE the += 2
        b[i] = b[i + 1] - b[i - 1];
    }
    return b;
}

public static int[] delta(int[] a) {
    if (a.length == 0) {
        return new int[0];
    }

    int[] b = new int[a.length + (a.length - 1)];
    for (int i = 0; i < a.length - 1; i++) {
        b[2 * i] = a[i];
        b[2 * i + 1] = a[i + 1] - a[i];
    }
    b[b.length - 1] = a[a.length - 1];
    return b;
}

public static int[] delta(int[] a) {
    if (a.length == 0) {
        return new int[0];
    }

    int[] result = new int[a.length + a.length - 1];
    int place = 0;
    for (int i = 0; i <= result.length - 1; i++) {
        if (i % 2 == 0) {
            result[i] = a[place];
            place++;
        } else {
            result[i] = a[place] - a[place - 1];
        }
    }
    return result;
}
```

10. Three possible solutions appear below.

```
public static void printReversed(String str) {
    for (int i = 0; i < str.length(); i++) {
        if (str.charAt(i) == ' ') {
            System.out.print(" ");
        } else {
            int j = i;
            while (j < str.length() && str.charAt(j) != ' ') {
                j++;
            }
            for (int k = j - 1; k >= i; k--) {
                System.out.print(str.charAt(k));
            }
            i = j;
        }
    }
    System.out.println();
}
```

```
public static void printReversed(String str) {
    Scanner input = new Scanner(str);
    for (int i = 0; i < str.length(); i++) {
        if (str.charAt(i) == ' ') {
            System.out.print(" ");
        } else {
            String word = input.next();
            for (int j = word.length() - 1; j >= 0; j--) {
                System.out.print(word.charAt(j));
            }
            i += word.length() - 1;
        }
    }
    System.out.println();
}
```

```
public static void printReversed(String str) {
    String build = "";
    for (int i = 0; i < str.length(); i++) {
        if (str.charAt(i) == ' ') {
            if (build.length() != 0) {
                System.out.print(build);
                build = "";
            }
            System.out.print(" ");
        } else {
            build = str.charAt(i) + build;
        }
    }
}
```