Key to CSE142 Final, Summer 2016

1.

Original List	Final List
{2, 5, 8}	{2, 5, 10}
{3, 4, 7, 5}	{3, 4, 10, 5}
{2, 4, 6, 2, 6}	{2, 4, 8, 2, 14}
{4, 5, 8, 9, 3}	{4, 5, 12, 9, 15}
{1, 2, 8, 5, 10, 5, 4}	{1, 2, 9, 5, 19, 5, 23}

2. Inheritance. The output produced is as follows.

```
retriever   retriever      dog      retriever
dog 1       dog 1          dog 1    labrador 1
golden 2    retriever 2    dog 2    retriever 2
```

3. Line-Based File Processing. One possible solution appears below.

```
    public static void underline(Scanner input) {
        while (input.hasNextLine()) {
            String text = input.nextLine();
            if (!text.startsWith(".")) {
                System.out.println(text);
            } else {
                System.out.println(text.substring(1));
                for (int i = 1; i <= text.length() - 1; i++) {
                    System.out.print("-");
                }
                System.out.println();
            }
        }
    }
```

4. Arrays. Two possible solutions appear below.

```
    public static boolean sameGap(int[] list) {
        if (list.length >= 2) {
            int gap = Math.abs(list[0] - list[1]);
            for (int i = 2; i < list.length; i++) {
                int gap2 = Math.abs(list[i] - list[i - 1]);
                if (gap != gap2) {
                    return false;
                }
            }
        }
        return true;
    }

    public static boolean sameGap(int[] list) {
        for (int i = 0; i < list.length - 2; i++) {
            int gap1 = Math.abs(list[i] - list[i + 1]);
            int gap2 = Math.abs(list[i + 1] - list[i + 2]);
            if (gap1 != gap2) {
                return false;
            }
        }
        return true;
    }
```

5. Critters. One possible solution appears below.

```
public class Eagle extends Critter {
    private int count;
    private int max;

    public Eagle() {
        count = 0;
        max = 1;
    }

    public Action getMove(CritterInfo info) {
        count++;
        if (count == 2 * max) {
            count = 0;
            max++;
        }
        if (info.getFront() == Neighbor.EMPTY) {
            return Action.HOP;
        } else if (info.getFront() == Neighbor.WALL) {
            return Action.RIGHT;
        } else {
            return Action.INFECT;
        }
    }

    public Color getColor() {
        if (count < max) {
            return Color.RED;
        } else {
            return Color.BLUE;
        }
    }

    public String toString() {
        return "<>";
    }
}
```

6. Art bonus

7. Reference Mystery. The program produces the following output.

```
1  7
-1 [2, 4, 6]
5  [2, 4, 6]
```

8. Token-Based File Processing. One possible solution appears below.

```
public static double evaluate(Scanner input) {
    double result = input.nextDouble();
    while (input.hasNext()) {
        String operator = input.next();
        double num = input.nextDouble();
        if (operator.equals("+")) {
            result += num;
        } else {  // operator.equals("-")
            result -= num;
        }
    }
    return result;
}
```

9. ArrayLists. Two possible solutions appear below.

```
public static void split(ArrayList<Integer> list) {
    for (int i = 0; i < list.size(); i += 2) {
        int n = list.get(i);
        list.set(i, n / 2 + n % 2);
        list.add(i + 1, n / 2);
    }
}

public static void split(ArrayList<Integer> list) {
    for (int i = 0; i < list.size(); i += 2) {
        int n = list.remove(i);
        list.add(i, n / 2 + n % 2);
        list.add(i + 1, n / 2);
    }
}
```

10. Arrays. One possible solution appears below.

```
public static int[] collapse(int[] list) {
    int[] result = new int[list.length / 2 + list.length % 2];
    for (int i = 0; i < list.length; i++)
        result[i / 2] += list[i];
    return result;
}
```

11. Programming. One possible solution appears below.

```
public static String encode(String s, int n) {
    String result = "";
        for (int j = 0; j < n; j++) {
            for (int i = 0; i < s.length() - j; i += n) {
                result += s.charAt(i + j);
            }
        }
    return result;
}
```