Sample Final Exam #1 Key

1.

Call	Final Contents of Array

`int[] a1 = {7};`
`arrayMystery(a1);`
	`{7}`

`int[] a2 = {4, 3, 6};`
`arrayMystery(a2);`
	`{4, 2, -2}`

`int[] a3 = {7, 4, 8, 6, 2};`
`arrayMystery(a3);`
	`{7, 4, -2, -5, -3}`

`int[] a4 = {10, 2, 5, 10};`
`arrayMystery(a4);`
	`{10, 9, 6, -1}`

`int[] a5 = {2, 4, -1, 6, -2, 8};`
`arrayMystery(a5);`
	`{2, -1, 2, -1, 5, 2}`

2.

```
14 14
7 9 14 2
18 18
7 9 14 18
```

3.

```
b
c 1
a 2  c 1

b
c 1
b 2  c 2

c
c 1
c 2

b
d 1  b 2  c 2
b 2  c 2
```

4.

```
public static void printStrings(Scanner input) {
    while (input.hasNextInt()) {
        int times = input.nextInt();
        String word = input.next();
        for (int i = 0; i < times; i++) {
            System.out.print(word);
        }
        System.out.println();
    }
}
```

5.

```
public static void reverseLines(Scanner input) {
    while (input.hasNextLine()) {
        String text = input.nextLine();
        for (int i = text.length() - 1; i >= 0; i--) {
            System.out.print(text.charAt(i));
        }
        System.out.println();
    }
}
```

6.

```
public static boolean isAllEven(int[] list) {
    for (int i = 0; i < list.length; i++) {
        if (list[i] % 2 != 0) {
            return false;
        }
    }
    return true;
}
```

7.

```
public static boolean isUnique(int[] list) {
    for (int i = 0; i < list.length; i++) {
        for (int j = i + 1; j < list.length; j++) {
            if (list[i] == list[j]) {
                return false;
            }
        }
    }
    return true;
}
```

8.

```
import java.awt.*;    // for Color
import java.util.*;   // for Random

public class Ostrich extends Critter {
    private Random rand;
    private int steps;
    private boolean west;     // true if going west; false if east
    private boolean hiding;

    public Ostrich() {
        rand = new Random();
        hiding = true;
        steps = 0;
        west = rand.nextBoolean();   // or call nextInt(2) and map 0=false, 1=true
    }

    public Color getColor() {
        if (hiding) {
            return Color.CYAN;
        } else {
            return Color.WHITE;
        }
    }

    public Direction getMove() {
        if (steps == 10) {
            steps = 0;   // Pick a new direction and re-set the steps counter
            hiding = !hiding;
            west = rand.nextBoolean();
        }

        steps++;
        if (hiding) {
            return Direction.CENTER;
        } else if (west) {
            return Direction.WEST;
        } else {
            return Direction.EAST;
        }
    }
}
```

9. Two solutions are shown.

```
public int compareTo(Date other) {
    if (month < other.month || (month == other.month && day < other.day)) {
        return -1;
    } else if (month == other.month && day == other.day) {
        return 0;
    } else {
        return 1;
    }
}

public int compareTo(Date other) {
    if (month == other.month) {
        return day - other.day;
    } else {
        return month - other.month;
    }
}
```