

CS 142 Sample Midterm Exam #1 Key

1. Array Mystery

```
{1}
{2, 7, 9}
{2, 5, 2, 2}
{2, 6, -1, 3, 3}
{5, 6, 2, 6, 2, 4}
```

2. ArrayList Mystery

```
{34, 20, 2}
{12, 20, 40, 34, 1}
{54, 20, 40, 60, 7, 4}
```

3. Complexity

```
n^2
n
n^2
n^2
```

4. Polymorphism

```
Frodo 1/Bilbo 1
Bilbo 1
Gandalf 1
Bilbo 1
Frodo 1/Bilbo 1
Gandalf 1
Gandalf 2/Frodo 1/Bilbo 1
Gandalf 2/Bilbo 1
Gandalf 2/Gandalf 1
Gandalf 2/Bilbo 1
Gandalf 2/Frodo 1/Bilbo 1
Gandalf 2/Gandalf 1
compiler error
Gandalf 2/Frodo 1/Bilbo 1
runtime error
Gandalf 2/Gandalf 1
Bilbo 1
runtime error
runtime error
Frodo 3
```

5. Array Programming

Two possible solutions appear below.

```
public static void removeZeros(int[] list) {
    for (int i = list.length - 2; i >= 0; i--) {
        if (list[i] == 0) {
            for (int j = i; j < list.length - 1; j++) {
                list[j] = list[j + 1];
            }
            list[list.length - 1] = 0;
        }
    }
}
```

```

public static void removeZeros(int[] list) {
    int numNonZero = 0;
    for (int i = 0; i < list.length; i++) {
        if (list[i] != 0) {
            list[numNonZero] = list[i];
            numNonZero++;
        }
    }
    for (int i = numNonZero; i < list.length; i++) {
        list[i] = 0;
    }
}

```

6. ArrayList Programming

```

public static void removeAdjacentMatches(ArrayList<Integer> list) {
    for (int i = 0; i < list.size() - 1; i++) {
        if (list.get(i) == list.get(i + 1)) {
            list.remove(i + 1);
            i--;
        }
    }
}

```

7. Critter Programming

```

public class Eagle extends Critter {
    private int count;
    private int max;
    boolean fought;

    public Eagle() {
        count = 0;
        max = 1;
        fought = false;
    }
    public Direction getMove() {
        count++;
        if (count == 2 * max) {
            count = 0;
            max++;
        }
        if(fought) {
            return Direction.East;
        } else {
            return Direction.North;
        }
    }
    public Color getColor() {
        if (count < max) {
            return Color.RED;
        } else {
            return Color.BLUE;
        }
    }
    public String toString() {
        return "<>";
    }
    public Attack fight() {
        fought = true;
        return Attack.Pounce;
    }
}

```