Spring 2024 Exam 1 Answer Key

Problem 4:

```
public class Outfit {
    public static void main(String[] args) {
        // starter arrays with individual clothes
        String[] bhrajitShirts = { "yellow", "black", "button down brown", "denim" };
        String[] priyaShirts = { "red", "sparkly mesh", "black tank", "fruits", "purple" };

        // combined array initialization
        String[] combinedWardrobe = new String[bhrajitShirts.length + priyaShirts.length];
        for (int i = 0; i < combinedWardrobe.length; i++) {
            // fill up array first with Bhrajit's clothes
            if (i < bhrajitShirts.length) {
                combinedWardrobe[i] = bhrajitShirts[i];
                // after Bhrajit's clothes, fill up with Priya's clothes
            } else {
                combinedWardrobe[i] = priyaShirts[i - bhrajitShirts.length];
            }
        }
        // Select ANY valid index from the combinedWardrobe array twice.
        int bhrajitIndex = (int) (Math.random() * (combinedWardrobe.length));
        int priyaIndex = (int) (Math.random() * (combinedWardrobe.length));
        if (priyaIndex == bhrajitIndex) { // check if both selected the same index
            priyaIndex = (priyaIndex + 1) % combinedWardrobe.length; // if so, pick the next index
        }
        String bhrajitFit = combinedWardrobe[bhrajitIndex];
        String priyaFit = combinedWardrobe[priyaIndex];

        System.out.println(bhrajitFit);
        System.out.println(priyaFit);
    }
}
```

Problem 5:

```
public class HikePlanning {
    public static boolean isExciting(double[] hike) {
        double max = hike[0];
        double firstElevation = hike[0];
        for (int i = 1; i < hike.length; i++) {
            if (hike[i] > max) {
                max = hike[i];
            }
        }
        return max >= firstElevation * 2;
    }

    public static int firstDramaticSegment(double[] hike) {
        double current = hike[0];
        for (int i = 1; i < hike.length; i++) {
            if (Math.abs(hike[i] - current) > 1000) {
                return i - 1;
            }
            current = hike[i];
        }
        return -1;
    }

    public static boolean likeHike(double[] hike) {
        return isExciting(hike) && firstDramaticSegment(password) >= 1;
    }
}
```