CSE 122 Practice Final Solutions
Spring 2023

1. Conceptual
   a. Option A, Option C
   b. Option B
   c. One possible answer is shown for each
      i. {hello=hello}
      ii. {word=AnotherWord}

2. Code Tracing:
   a. {brick, plaster}
   b. {blue, green, yellow}
   c. {fruit}
   d. {animal, cat, dog, ipl}

3. Debugging:
   a. 7
   b. One possible solution is shown below.

      Delete line 7 and replace it with the following code put inside the body of the first loop

```
if (counts.containsKey(word.length()) {
    counts.put(word.length(), counts.get(word.length()) + 1);
else {
    counts.put(word.length(), 1);
}
```

4. Collections Programming: One possible solution is shown below

```
public void removeShorterStrings(List<String> list) {
    for (int i = 0; i < list.size() - 1; i++) {
        String first = list.get(i);
        String second = list.get(i + 1);
        if (first.length() <= second.length()) {
            list.remove(i);
        } else {
            list.remove(i + 1);
        }
    }
}
```

5. Objects Programming: One possible solution is shown below

```
public class GeneralItem implements ItemListing {

    private String name;
    private String brand;
    private double reviewTotal;
    private int numReviews;

    public GeneralItem(String name, String brand) {
        this.name = name;
        this.brand = brand;
    }

    public String getName() {
        return name;
    }

    public String getBrand() {
        return brand;
    }
```

```
    public void review(double rating) {
        reviewTotal += rating;
        numReviews++;
    }

    public double getRating() {
        if (numReviews == 0) {
            return 0.0;
        else {
            return reviewTotal / numReviews;
        }
    }

    public String toString() {
        String result = name + " (" + brand + ") rates " + getRating() + " (";
        if (numReviews == 1) {
            result += "1 review)";
        else {
            result += numReviews + " reviews)";
        }
        return result;
    }

    public boolean isDuplicate(ItemListing other) {
        return other.getName().equals(name) && other.getBrand().equals(brand);
    }
```

6. Stacks & Queues: One possible solution appears below.

```
    public void reorder(Queue<Integer> q) {
        Stack<Integer> s = new Stack<>();
        int oldSize = q.size();
        for (int i = 0; i < oldSize; i++) {
            int n = q.remove();
            if (n < 0) {
                s.push(n);
            } else {
                q.add(n);
            }
        }
        int newSize = q.size();
        while (!s.isEmpty()) {
            q.add(s.pop();
        }
        for (int i = 0; i < newSize; i++) {
            q.add(q.remove());
        }
    }
```