CSE115 Introduction to Computer Science I
Quiz #3 Retrospective — Fall 2017

In this quiz we asked you to study some code and then answer some questions. The given code was:

```
public ArrayList<String> mystery(ArrayList<String> a) {
        ArrayList<String> c = new ArrayList<String>();
        for (int i = 1; i < a.size() ; i = i + 2) {
                c.add(a.get(i));
                c.add(a.get(i-1));
        }
        return c;
}

public void question1() {
        ArrayList<String> listA = new ArrayList<String>();
        listA.add("Fry");
        listA.add("Leela");
        listA.add("Bender");
        listA.add("Zoidberg");
        System.out.println( mystery(listA) );
}

public void question2() {
        ArrayList<String> listB = new ArrayList<String>();
        listB.add("Echo");
        listB.add("Topher");
        listB.add("Victor");
        listB.add("Boyd");
        listB.add("Whiskey");
        System.out.println( mystery(listB) );
}

public void question3() {
        ArrayList<String> listC = new ArrayList<String>();
        listC.add("Denny");
        System.out.println( mystery(listC));
}
```

QUESTION 1: *What is printed when question1() is called?*

To answer this question we need to look at the definition of the method question1():

```
public void question1() {
        ArrayList<String> listA = new ArrayList<String>();
        listA.add("Fry");
        listA.add("Leela");
        listA.add("Bender");
        listA.add("Zoidberg");
        System.out.println( mystery(listA) );
```

```
	}
```

This method creates an ArrayList<String> and populates it with the Strings “Fry”, “Leela”, “Bender” and “Zoidberg”, calls the mystery method with this last as an argument, and then prints the value that mystery returns. Mystery is defined as follows:

```
	public ArrayList<String> mystery(ArrayList<String> a) {
		ArrayList<String> c = new ArrayList<String>();
		for (int i = 1; i < a.size() ; i = i + 2) {
			c.add(a.get(i));
			c.add(a.get(i-1));
		}
		return c;
	}
```

This method creates a new ArrayList<String> and inserts into that new ArrayList elements from its input ArrayList, but in a different order. With the ArrayList [“Fry”, “Leela”, “Bender”, “Zoidberg”] as input, this is how the mystery method operates:

```
AT START OF METHOD:
    a is: [Fry, Leela, Bender, Zoidberg]
    c is: []
    AT START OF LOOP BODY, i has value: 1
        a.get(1) has value: Leela
        after adding this value, c is: [Leela]
        a.get(0) has value: Fry
        after adding this value, c is: [Leela, Fry]
    AT START OF LOOP BODY, i has value: 3
        a.get(3) has value: Zoidberg
        after adding this value, c is: [Leela, Fry, Zoidberg]
        a.get(2) has value: Bender
        after adding this value, c is: [Leela, Fry, Zoidberg, Bender]
```

The value returned and printed is:

[Leela, Fry, Zoidberg, Bender]

QUESTION 2: *What is printed when question2() is called?*

We reason as we did for question 1. A different ArrayList is created, which is then passed as an argument to mystery:

```
public void question2() {
        ArrayList<String> listB = new ArrayList<String>();
        listB.add("Echo");
        listB.add("Topher");
        listB.add("Victor");
        listB.add("Boyd");
        listB.add("Whiskey");
        System.out.println( mystery(listB) );
}
```

With the ArrayList [“Echo”, “Topher”, “Victor”, “Boyd”, “Whiskey”] as input, this is how the mystery method operates:

```
AT START OF METHOD:
    a is: [Echo, Topher, Victor, Boyd, Whiskey]
    c is: []
    AT START OF LOOP BODY, i has value: 1
        a.get(1) has value: Topher
        after adding this value, c is: [Topher]
        a.get(0) has value: Echo
        after adding this value, c is: [Topher, Echo]
    AT START OF LOOP BODY, i has value: 3
        a.get(3) has value: Boyd
        after adding this value, c is: [Topher, Echo, Boyd]
        a.get(2) has value: Victor
        after adding this value, c is: [Topher, Echo, Boyd, Victor]
```

The value returned and printed is:

```
[Topher, Echo, Boyd, Victor]
```

QUESTION 3: *What is printed when question3() is called?*

We reason as we did in questions 1 and 2. A different ArrayList is created, which is then passed as an argument to mystery:

```
public void question3() {
      ArrayList<String> listC = new ArrayList<String>();
      listC.add("Denny");
      System.out.println( mystery(listC));
}
```

With the ArrayList [“Denny”] as input, this is how the mystery method operates:

```
AT START OF METHOD:
    a is: [Denny]
    c is: []
```

The value returned and printed is:

```
[]
```