Building Java Programs

Reference Semantics

What does this do?

```
public class ReferenceMystery {
    public static void main(String[] args) {
        ArrayList<String> food = new ArrayList<String>();
        food.add("cookies");
        food.add("milk");
        food.add("pizza");
        food.add("ice cream");
        food.add("tomato");


        ArrayList<String> food2 = food;
        removeSomeFood(food2);

        System.out.println("food: " + food);
        System.out.println("food 2: " + food2);
    }

    public static void removeSomeFood(ArrayList<String> food) {
        for (int i = 0; i < food.size(); i++) {
            if (food.get(i).contains("c")) {
                food.remove(i);
                i--;
            }
        }
    }
}
```

Reference Mystery Answer

Output:

```
food: [milk, pizza, tomato]
food 2: [milk, pizza, tomato]
```

- Creating a reference:

```
ArrayList<String> food2 = food;
```

  - When creating a variable that references another object, you are storing a copy of the reference to the object NOT a copy of the object itself
  - If you modify the object by using one of the references, the effect is seen through both of the references
  - Only a call on `new` can create a new object

House Analogy

- A house object is:
  - built on a plot of land
  - referenced by its address
  - contains some number of rooms

- What do we need to keep track of inside the House object?

- `var1 = new House()` **builds a house**
  - makes a new plot of land for the house

- `var1 = var2` **makes the variables refer to the same address**
  - does not make a new plot of land and does not make a new house

House Analogy

```
House h1 = new House();
House h2 = new House();
House h3 = h2;
```

- Are any of these houses on the same plot of land or have the same address?
  - No, there are two plots of land: one for h1 and one for h2
  - Yes, h2 and h3 have the same address, they are the same house.
- If h1 and h2 are decorated exactly the same, are they the same house?
  - No, but they could be considered .equals()