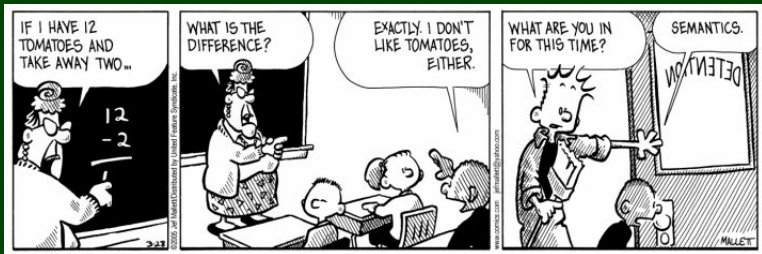


CSE 143

Computer Programming II

Reference Semantics



See Piazza.

```
1 public static void listMangler(ArrayList<String> list) {
2     for (int i = 0; i < list.size(); i++) {
3         list.set(i, "");
4     }
5 }
6
7 public static void main(String[] args) {
8     ArrayList<String> animals = new ArrayList<String>();
9     animals.add("Elephant");
10    animals.add("Bunny");
11    animals.add("Zebra");
12
13
14    ArrayList<String> animals2 = animals;
15
16    listMangler(animals);
17
18    System.out.println(animals);
19    System.out.println(animals2);
20 }
```

OUTPUT

```
>> [, , ]
>> [, , ]
```

What Are We Doing...?

We're trying to understand how Java passes arguments to our methods.

Today's Main Goals:

- To understand how Java passes arguments to methods
- To understand what `null` is
- To understand the difference between primitive and Object types

Talk to the person next to you and try to answer the following two questions: Unambiguously describe. . .

- . . . the day of the month you were born.
- . . . every detail of the house you grew up in.

Your descriptions should be good enough that the person next to you could (given enough time) **completely recreate** the answer.

Bottom Line: The first one is really easy; the second one is hard.

What's a better way of describing the house?

Give your address which allows the person to look at the house!

A **house** is:

- Built on a **plot of land**
- Identified by its **address**
- Made up of **rooms** which are decorated a **color**

Java Lane

100:

yellow

bdrm[0]

200:

yellow

bdrm[0]

300:

blue

red

bdrm[0]

bdrm[1]

Some Questions

- Are any of these houses on the same piece of land?
- What if 100 Java Lane and 200 Java Lane are decorated identically?
They're kind of the same now, right?

```
1 public class Room {
2     // BAD STYLE (public field)
3     String decorations;
4 }
5 public class House {
6     // BAD STYLE (public field)
7     Room[] bedrooms;
8 }
```

Suppose House `h100` refers to “100 Java Lane”. What does the following code do?

```
House hx = h100;
```

It doesn't actually change the plots!

The statement makes `h100` refer to the same house as `hx`, which is “100 Java Lane”.

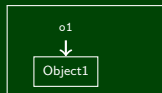
`var1 = new Object()` **builds a house.**

`var2 = var1` **makes the variables refer to the same address.**

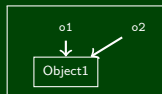
Draw a picture of the “street” after each line of this code. How many houses were built? How many addresses were given out?

```
1 Object o1 = new Object();  
2 Object o2 = o1;  
3 Object o3 = new Object();  
4 o3 = o2;
```

```
Object o1 = new Object();
```



```
Object o2 = o1;
```



```
Object o3 = new Object();
```



```
o3 = o2;
```



```
1 public class Room {
2     String decorations;
3 }
4 public class House {
5     Room[] bedrooms;
6 }
```

We want to make a list of the color of every bedroom on Java Lane.
Write down a procedure for doing this.

- Go to a house
- Go to each bedroom in the house and write down its color

In code:

```
1 public void decorations(House h) {
2     int num = h.bedrooms.length;
3     for (int i = 0; i < h.bedrooms.length; i++) {
4         System.out.println(h.bedrooms[i].decorations);
5     }
6 }
```

In Java, the “.” means “use the address to locate the thing”

```
1 public class Room {
2     String decorations;
3 }
4 public class House {
5     Room[] bedrooms;
6 }

1 Room myRoom = new Room();
2 myRoom.decorations = "black";
3
4 Room[] rooms = new Room[1];
5 room[0] = myRoom;
6
7 House h = new House();
8 h.bedrooms = rooms;
9
10 /* This method should repaint all the rooms of
11    * the specified house to be green. */
12 public static void repaintHouse(House house) {
13 }
```

```
1 public class Room {
2     String decorations;
3 }
4 public class House {
5     Room[] bedrooms;
6 }

1 Room myRoom = new Room();
2 myRoom.decorations = "black";
3
4 Room[] rooms = new Room[1];
5 //room[0] = myRoom;
6
7 House h = new House();
8 h.bedrooms = rooms;
9
10 /* This method should repaint all the rooms of
11    * the specified house to be green. */
12 public static void repaintHouse(House house) {
13 }
```

```
1 public class Room {
2     String decorations;
3 }
4 public class House {
5     Room[] bedrooms;
6 }

1 Room myRoom = new Room();
2 myRoom.decorations = "black";
3
4 Room[] rooms = new Room[1];
5 /* This is like saying the house was supposed to have a
6  * bedroom, but it was never made. */
7 room[0] = null;
8
9 House h = new House();
10 h.bedrooms = rooms;
11
12 /* This method should repaint all the rooms of
13  * the specified house to be green. */
14 public static void repaintHouse(House house) {
15 }
```

Consider the following code:

```
ArrayList<Integer> list = new ArrayList<Integer>();
```

Q: The house is the `ArrayList`, what are the rooms?

Describe in the house analogy what each of the following lines of code do

```
1 ArrayList<Integer> list2 = list;
2 ArrayList<Integer> list3 = new ArrayList<Integer>();
3 list2.add(5);
4 list3.add(7);
5 append9000(list2);
6
7 public void append9000(ArrayList<Integer> list) {
8     list.add(9000);
9 }
```

```
1 public void youGuess(int theAnswer) {
2     theAnswer = 1000;
3     System.out.println("I guess " + theAnswer);
4 }
5
6 public static void main(String[] args) {
7     int myNumber = 42;
8     youGuess(myNumber);
9     System.out.println("The right answer is " + myNumber);
10 }
```



```
1 public class TheAnswer {
2     int answer;
3 }
4
5 public void youGuess(TheAnswer ans) {
6     ans.answer = 1000;
7     System.out.println("I guess " + ans.answer);
8 }
9
10 public static void main(String[] args) {
11     TheAnswer ans = new TheAnswer();
12     ans.answer = 42;
13     youGuess(ans);
14     System.out.println("The right answer is " + ans.answer);
15 }
```

Consider the following. It's your birthday, and...

- I ask you "how old were you before today?"
- You answer `prevYearsOld`
- In my head, I increment that value (`prevYearsOld += 1`).
- I shout out "You're `prevYearsOld` now!"

If I were to ask you again, would you answer differently?

```
1 public shoutOnBirthday(int prevYearsOld) {
2     prevYearsOld += 1;
3     System.out.println("You're " + prevYearsOld + " years old now!");
4 }
5
6 public static void main(String[] args) {
7     int prevYearsOld = 350; /* Loch Ness Monster */
8     shoutOnBirthday(prevYearsOld);
9 }
```

```
1 public void doIt(int[] array) {
2     if (array == null) {
3         array = new int[1];
4         array[0] = 9999;
5     }
6     else {
7         array[0] = 101;
8     }
9 }
10
11 public static void main(String[] args) {
12     int[] array = null;
13     doIt(null);
14     doIt(array);
15     doIt(array);
16 }
```