# CSE 143

## Computer Programming II

# Reference Semantics



## Questions & Drawings From Last Time 1

See Piazza.

## What Does This Do? 2

```
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 Again? 3

**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

## A Silly Question 4

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 | | 200: | yellow | | 300: | blue | red |
|------|--------|---|------|--------|---|------|------|-----|
| | bdrm[0] | | | bdrm[0] | | | 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;
```
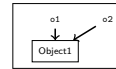
---

```
Object o1 = new Object();
```
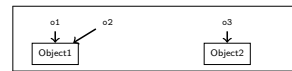
| o1 |
|----|
| ↓ |
| Object1 |

```
Object o2 = o1;
```

o1   o2
↓
Object1

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

o1   o2          o3
↓               ↓
Object1         Object2

```
o3 = o2;
```

o1/o2/o3
↓
Object1         Object2

---

```
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 }
```

```java
public class Room {
    String decorations;
}
public class House {
    Room[] bedrooms;
}
```

```java
Room myRoom = new Room();
myRoom.decorations = "black";

Room[] rooms = new Room[1];
//room[0] = myRoom;

House h = new House();
h.bedrooms = rooms;

/* This method should repaint all the rooms of
 * the specified house to be green. */
public static void repaintHouse(House house) {
}
```

```java
public class Room {
    String decorations;
}
public class House {
    Room[] bedrooms;
}
```

```java
Room myRoom = new Room();
myRoom.decorations = "black";

Room[] rooms = new Room[1];
/* This is like saying the house was supposed to have a
 * bedroom, but it was never made. */
room[0] = null;

House h = new House();
h.bedrooms = rooms;

/* This method should repaint all the rooms of
 * the specified house to be green. */
public static void repaintHouse(House house) {
}
```

Consider the following code:

```java
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

```java
ArrayList<Integer> list2 = list;
ArrayList<Integer> list3 = new ArrayList<Integer>();
list2.add(5);
list3.add(7);
append9000(list2);

public void append9000(ArrayList<Integer> list) {
    list.add(9000);
}
```

```java
public void youGuess(int theAnswer) {
    theAnswer = 1000;
    System.out.println("I guess " + theAnswer);
}

public static void main(String[] args) {
    int myNumber = 42;
    youGuess(myNumber);
    System.out.println("The right answer is " + myNumber);
}
```

```java
public class TheAnswer {
    int answer;
}

public void youGuess(TheAnswer ans) {
    ans.answer = 1000;
    System.out.println("I guess " + ans.answer);
}

public static void main(String[] args) {
    TheAnswer ans = new TheAnswer();
    ans.answer = 42;
    youGuess(ans);
    System.out.println("The right answer is " + ans.answer);
}
```

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?

```java
public shoutOnBirthday(int prevYearsOld) {
    prevYearsOld += 1;
    System.out.println("You're " + prevYearsOld + " years old now!");
}

public static void main(String[] args) {
    int prevYearsOld = 350; /* Loch Ness Monster */
    shoutOnBirthday(prevYearsOld);
}
```

```
 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  }
```