CSE 121 – Lesson 13

Justin Ung

Summer 2023

Music: 🌸k-pop girlies playlist🌸

No sli.do today <3

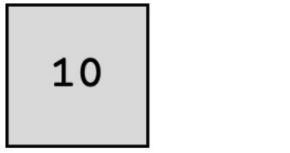
Announcements, Reminders

- Programming Assignment 3 Its Data Time released, due next Tuesday
- Resub 4 due yesterday, Resub 5 out now due next Thurs
- Quiz 2 (Take-home): Monday Aug 7th (8/7)
  - Topics: File I/O (Scanner, PrintStream), Arrays, Reference Semantics
- **Reminder:** Final exam Wednesday Aug 16 4:30 – 6:30 PM in PAA A102
  - Left-Handed Seating Request form posted on Ed due Aug 7th EOD

(PCM) Value Semantics vs. Reference Semantics

- Applies when working with primitive types
- Variables/parameters hold a *copy* of the actual value

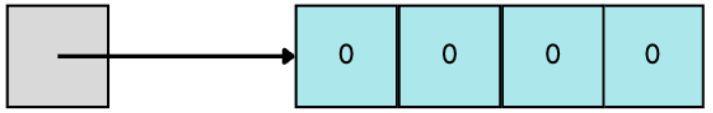
```
age = 10;
```



name: age (int)

- Applies when working with objects
- Variables/parameters hold a *reference* to the object

```
int[] arr = new int[4];
```

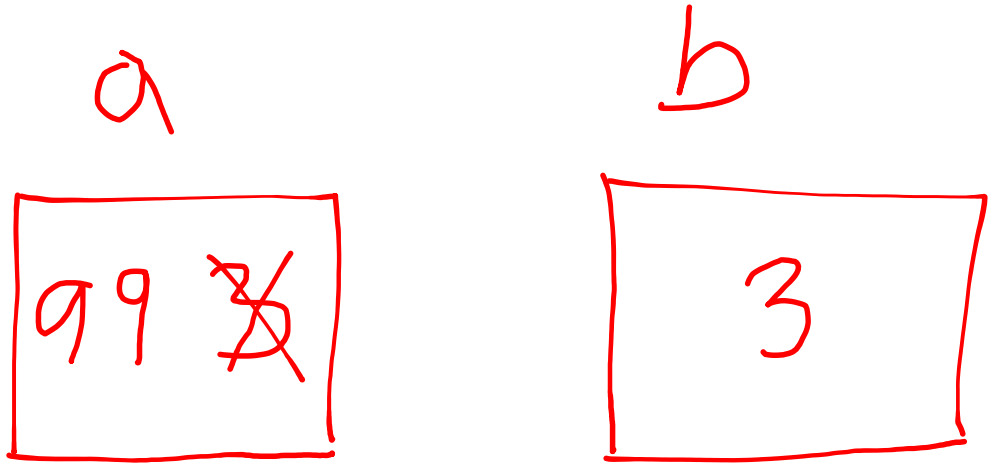


name: arr (int[])

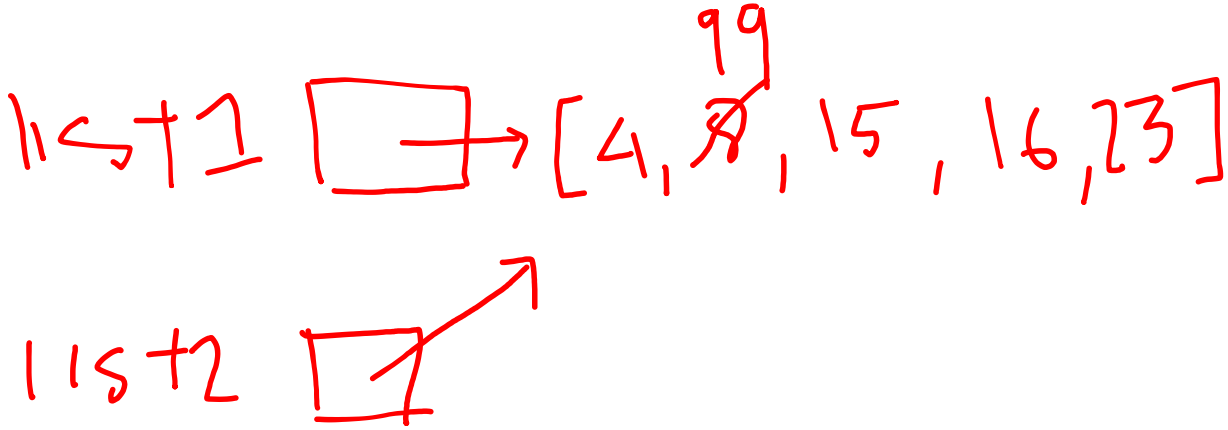
0 1 2 3

(PCM) Value Semantics vs. Reference Semantics

```
int a = 3;
int b = a;
a = 99;
```



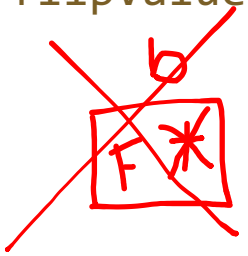

```
int[] list1 = {4, 8, 15, 16, 23};
int[] list2 = list1;
list1[1] = 99;
```



(PCM) Value Semantics vs. Reference Semantics

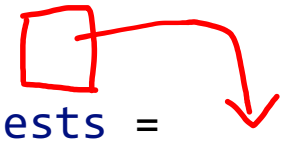
```
boolean test = true;
flipValue(test);

public static void flipValue(boolean b) {
    b = !b;
}
```




```
boolean[] tests =
        {true, true, false, true, false, false};
flipValues(tests);

public static void flipValues(boolean[] b) {
    for (int i = 0; i < b.length; i++) {
        b[i] = !b[i];
    }
}
```



Array Patterns!

(PCM) Counting Elements that Meet a Condition

"one"	"two"	"three"	"six"	"seven"	"eight"	"ten"

```
public static int evenLength(String[] list) {
    int countEven = 0;
    for (int i = 0; i < list.length; i++) {
        if (                          ) {
            countEven++;
        }
    }

    return countEven;
}
```

(PCM) Counting Elements that Meet a Condition

"one"	"two"	"three"	"six"	"seven"	"eight"	"ten"

```
public static int evenLength(String[] list) {
    int countEven = 0;
    for (int i = 0; i < list.length; i++) {
        if (list[i].length() % 2 == 0) {
            countEven++;
        }
    }

    return countEven;
}
```

(PCM) Modifying Elements of an Array

4	8	15	16	23	42

```
public static void clamp(int min, int max, int[] list) {
    for (int i = 0; i < list.length; i++) {
        if (                    > max) {
                               = max;
        } else if (                     < min) {
                               = min;
        }
    }
}
```

(PCM) Modifying Elements of an Array

4	8	15	16	23	42

```
public static void clamp(int min, int max, int[] list) {
    for (int i = 0; i < list.length; i++) {
        if (list[i] > max) {
            list[i] = max;
        } else if (list[i] < min) {
            list[i] = min;
        }
    }
}
```

(PCM) Searching for an Element

"one"	"two"	"three"	"six"	"seven"	"eight"	"ten"

```
public static int indexOfIgnoreCase(String phrase, String[] list) {
    for (int i = 0; i < list.length; i++) {
        if (                                        ) {
            return          ;
        }
    }

    return          ;
}
```

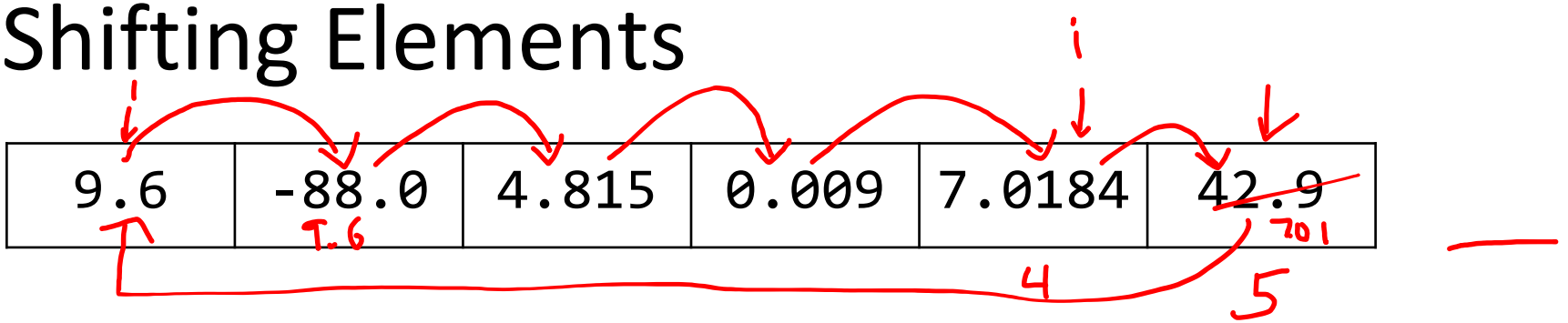
(PCM) Searching for an Element

"one"	"two"	"three"	"six"	"seven"	"eight"	"ten"

```
public static int indexOfIgnoreCase(String phrase, String[] list) {
    for (int i = 0; i < list.length; i++) {
        if (list[i].equalsIgnoreCase(phrase)) {
            return i;
        }
    }

    return -1;
}
```

(PCM) Shifting Elements




```
public static void rotateRight(double[] list) {
    double lastElement = list[list.length - 1];

    for (int i = list.length - 1; i > 0; i--) {
        list[i] = list[i - 1];
    }



}
```

(PCM) Shifting Elements

9.6	-88.0	4.815	0.009	7.0184	42.9

```
public static void rotateRight(double[] list) {
    double lastElement = list[list.length - 1];

    for (int i = list.length - 1; i > 0; i--) {
        list[i] = list[i - 1];
    }

    list[0] = lastElement;
}
```

(PCM) Looking at Multiple Elements in an Array

0	1	9	1	0

```
public static boolean isPalindrome(int[] list) {
    for (int i = 0; i < list.length / 2; i++) {
        if (list[i] != list[list.length - 1 - i]) {
            return             ;
        }
    }

    return             ;
}
```

(PCM) Looking at Multiple Elements in an Array

0	1	9	1	0

```
public static boolean isPalindrome(int[] list) {
    for (int i = 0; i < list.length / 2; i++) {
        if (list[i] != list[list.length - 1 - i]) {
            return false;
        }
    }

    return true;
}
```

(PCM) Array of Counters or "Tallying"

8	3	0	1	2	2	0	7	2

```
public static int[] numCount(Scanner input) {
    int[] counts =                  ;
    while (input.hasNextInt()) {
        int num = input.nextInt();



    }

    return counts;
}
```

(PCM) Array of Counters or "Tallying"

8	3	0	1	2	2	0	7	2

```
public static int[] numCount(Scanner input) {
    int[] counts = new int[10];
    while (input.hasNextInt()) {
        int num = input.nextInt();
        counts[num]++;
    }

    return counts;
}
```

(PCM) Common Ideas in Array Patterns

- Loop bounds
- Direction of traversal
- Indexing into an array