Lec 9

Arrays
Announcements

• Updated lab hours
How would we keep track of these 3 people?
What about 15 people?
CSE 11 comparison

• Write a program that keeps track of three student names input from the terminal
CSE 11 comparison

• Keep track of three student names input from the terminal

```java
String student1 = scnr.next();
String student2 = scnr.next();
String student3 = scnr.next();
```
CSE 11 comparison

• Write a program that keeps track of 15 student names input from the terminal
CSE 11 comparison

- Keep track of 15 student names input from the terminal

```java
String[] students = new String[15];
for(int i = 0; i < 15; i++) {
    students[i] = scnr.next();
}
```
int x;
x = 200;
int y = 100;
Array Memory

```c
int[] i = new int[3];
```
What gets printed?

```
int[] bikes = new int[5];
System.out.println(bikes);
```

A) 0 0 0 0 0
B) 0
C) [I@32x34 (memory address)
D) Unknown
E) Compiler error
Primitive Arrays - Syntax

```java
int[] GTAPeople = new int[3];  // 0 0 0

int numCars = 5;
float[] carSpeeds;
carSpeeds = new float[numCars];

double[] playerDamage = {3.23, 4.44};  // 3.23 4.44
```
Memory

• Primitive type variables are not initialized to 0
  – Compiler error if program uses variable before initialized
  – Primitive arrays are initialized to 0
    • What are char[], boolean[], int[], float[], double[] initialized to?

• Accessing out of array bounds leads to a run time error (array index out of bounds exception)
Array Indexing

• Indices start with 0
  – Think “number of elements before”

```java
int[] cards = new int[5];
cards[4] = 10;
int card5 = cards[4];
```

• Length of array (number of elements) is what you’d expect
  – cards.length
What gets printed

```java
int[] cards = new int[5];
cards[1] = 10;
int x = 4;
System.out.println(cards[0] + " " + cards[1] + " " + 
cards[x-3] + " " + cards.length);
```

A) 0 10 1 0  
B) 10 0 0 5  
C) Runtime error  
D) 0 10 10 5  
E) None of the above
What gets printed

```java
int[] cards = new int[5];
cards[1] = 10;
System.out.println(cards[0] + " " +
        cards[1] + " " + cards[cards.length]);
```

A) 0 10 5
B) 10 0 4
C) 0 10 0
D) Runtime error
E) Compiler error
Arrays and memory

```java
int x = 3;
int y = x;
System.out.println(y == 3); // true
x = 4;
System.out.println(y == 3); // true
```
Arrays and memory

```java
int[] x = {3};
int[] y = x;
System.out.println(y[0] == 3);
x[0] = 4;
System.out.println(y[0] == 3);
```
Arrays and memory

```java
int[] x = {3};
int[] y = x;
System.out.println(y[0] == 3);
x[0] = 4;
System.out.println(y[0] == 3);
```

![Diagram showing array memory allocation and pointer movement]
What gets printed

```java
int[] x = new int[3];int[] y = x;int i = 10;int j = i;
y[2] = -1;
i = 3;
System.out.println(x[2] + " " + j);
```

A) 0 10
B) -1 10
C) -1 3
D) 0 3
E) None of the above
public class Test {
    public static void newMethod(int[] i, int j) {
        i[0]++;
        j++;
        return;
    }

    public static void main(String[] args) {
        int[] i = {10, 20};
        newMethod(i, i[1]);
        System.out.println(i[0] + " " + i[1]);
    }
}

A) 10 20
B) 11 20
C) 10 21
D) 11 21
E) None