CSE 11
Accelerated Intro to Programming
Discussion Section 5

Sachin Deshpande,
Summer 1 2021

This discussion is being recorded

Logistics

- PA5 released yesterday, due Wednesday
- Exam THIS week!
- PA6 easier, PA7 hard.

just like programming assignments
Make a video

Be prepared
Come to us for help

Arrays

- An ordered sequence of values
  - (int, double, String, YourClass, array, ...)
- Syntax of creating an array:

```
int[] nums1 = { 4, 2, 7 };
int[] nums2 = {};
int[] nums3 = new int[5];
int[] nums4 = nums1;          // Be careful!!!
String[] strs1 = {"Hello", "CSE11"};
```

- Array length:

```
int nums1Length = nums1.length;
```

strs1.length; // 2

strs1[0].length(); // 5

Arrays cont'd

- Use index to access elements in an array:

```
int[] nums1 = { 4, 2, 7 };
int firstElem  = nums1[0];   // 4
int secondElem = nums1[1];   // 2
int thirdElem  = nums1[2];   // 7
int fourthElem = nums1[3];   // Index 3 out of bounds for length 3
```

// String[] nums1 = { "a", "b", "c" }

index starts with 0 !!!

- Modify elements in an array:

```
nums1[2] += 1;  // nums1 becomes { 4, 2, 8 }
```

1. → equivalent to → 2. nums1[2] = nums1[2] + 1;

→ incrementing nums1[2] by 1

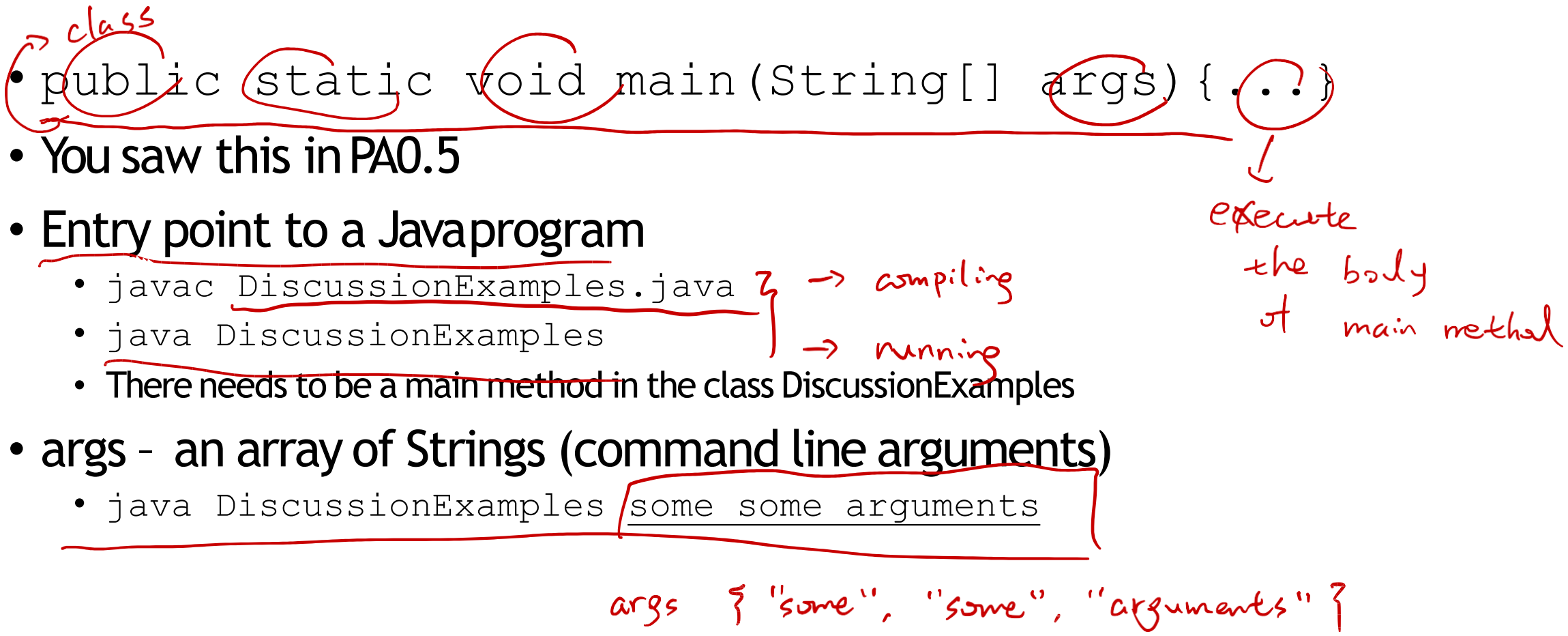
nums1[2] = 8;

3. nums1[2]++;

- Array length is fixed:
  - Can modify elements, but cannot add/delete elements
  - `nums1 = { 4, 2, 7, 8 } // need to assign a different array`

Main

- `public static void main(String[] args){...}`
- You saw this in PA0.5
- Entry point to a Java program
  - `javac DiscussionExamples.java` → compiling
  - `java DiscussionExamples` → running
  - There needs to be a main method in the class DiscussionExamples
- args - an array of Strings (command line arguments)
  - `java DiscussionExamples some some arguments`



Loop Basics

- How to access elements in a very very long array?
- Loops
  - For-loop

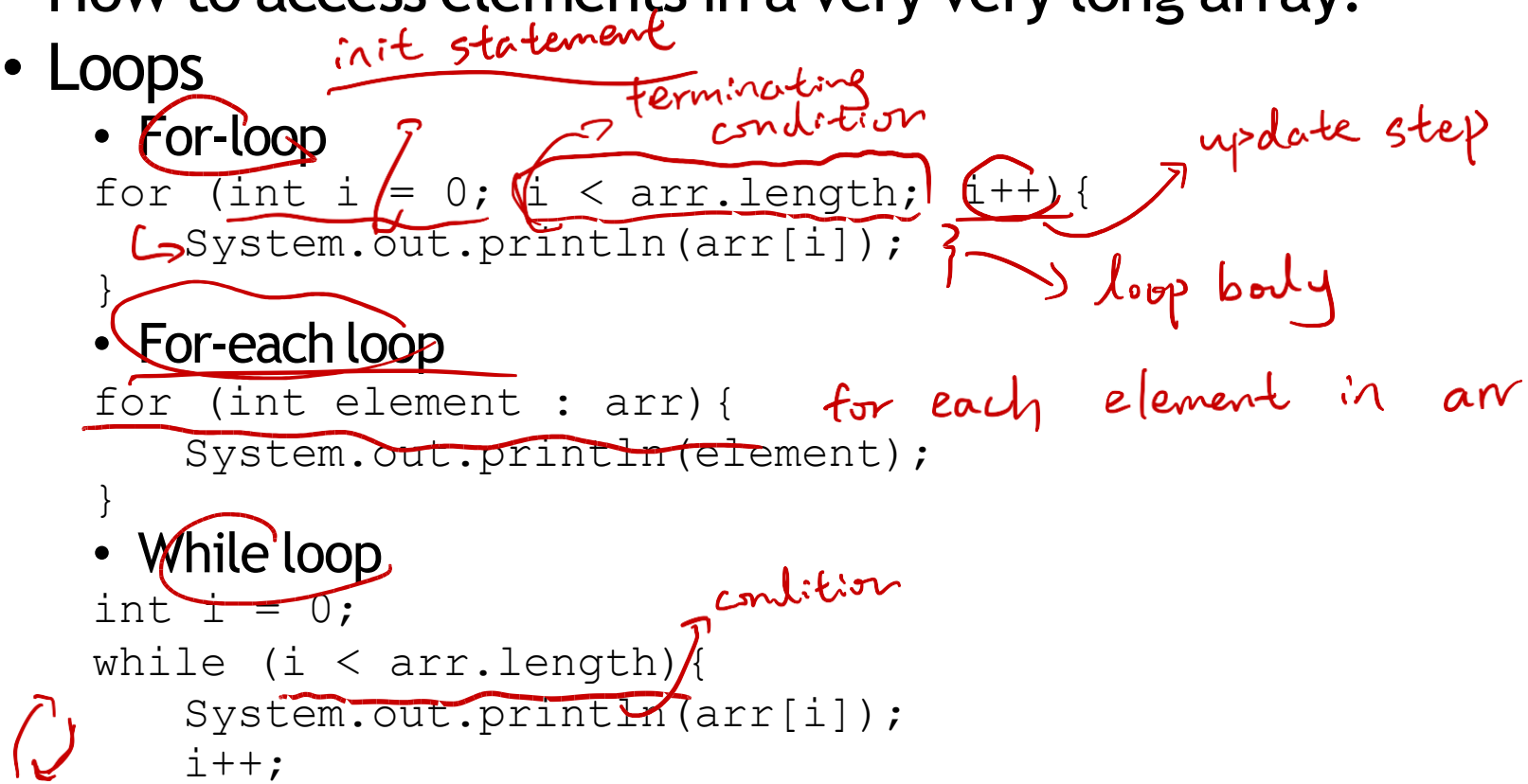
```
for (int i = 0; i < arr.length; i++){
    System.out.println(arr[i]);
}
```

  - For-each loop

```
for (int element : arr){
    System.out.println(element);
}
```

  - While loop

```
int i = 0;
while (i < arr.length){
    System.out.println(arr[i]);
    i++;
}
```



Quick Midterm Review…

Exam

Reserve enough time

Programming assignment: Coding + Video

Materials: Review lectures / readings / quizzes / PAs / discussions

Class — abstract classes
- interfaces
- normal classes

fields — types — primitive types { int, double, boolean
- classes { String, Pair, Point, ...

objects (new)

methods — constructors
- general method (ReturnType functionName (ArgType arg1, ...) { })
- main method

tester (t.checkExpect)

import tester.*; (./run)

if / else if / else

for-loops

String methods

Math methods

Not guaranteed to be complete

Fields vs Member Variables

Fields = member variables = instance variable
Variables in functions are different! Local variables

Tester

```
import tester.*;

class Test1 {
        int method1() {  . . . }
        // week1 - test as a single method to a field
        int field1 = this.method1();

       //wed of Week2
       boolean testMethod1(Tester t) {
              return t.checkExpects(this.method1(), 1);
       }
}
```

How it works:

- Method 1: observe output, see if it is what you expect
- Method 2: use a built-in method in Tester class to compare actual to expected value.

Testing

The point of testing is to break your program

Suppose I have a method foo(n) that adds 2 to your number
Would it be better to test foo(2), foo(3), foo(4),…?

No! Try foo(0), foo(-100000), foo(10000000), foo(“string”), etc.

PA5

"a" "abc"

1 2

Thanks!

"a".compareTo("abc") → negative

1 ~ 2 → negative