

CSC 212

PROGRAMMING WITH

DATA STRUCTURES

SPRING 2016

PROF. SARA SHEEHAN

SMITH COLLEGE

CLASS 9: FEB 23

OUTLINE

- **Midterm overview**
- **Demo Homework 3**
- **Debrief Lab 4**
- **Generics**
- **Java tip: assert**
- **Copying a data structure**
- **Begin 3rd data structure: Stacks**
- **MSA (mid-semester assessment) at 10am**

MIDTERM INFO

- **During LAB on Thursday March 10**
- **2 rooms based on account name**
- **1 double-sided handwritten note sheet**
- **Some practice midterms will be available**

MIDTERM INFO

Material

- Arrays
- Lists
- Stacks
- Queues

- Java fundamentals (classes, fields, methods, etc)
- Keywords and vocab (public/private, static, final, void, etc)
- Inheritance and Interfaces
- Runtime analysis
- Sorting
- Generics
- Iterators

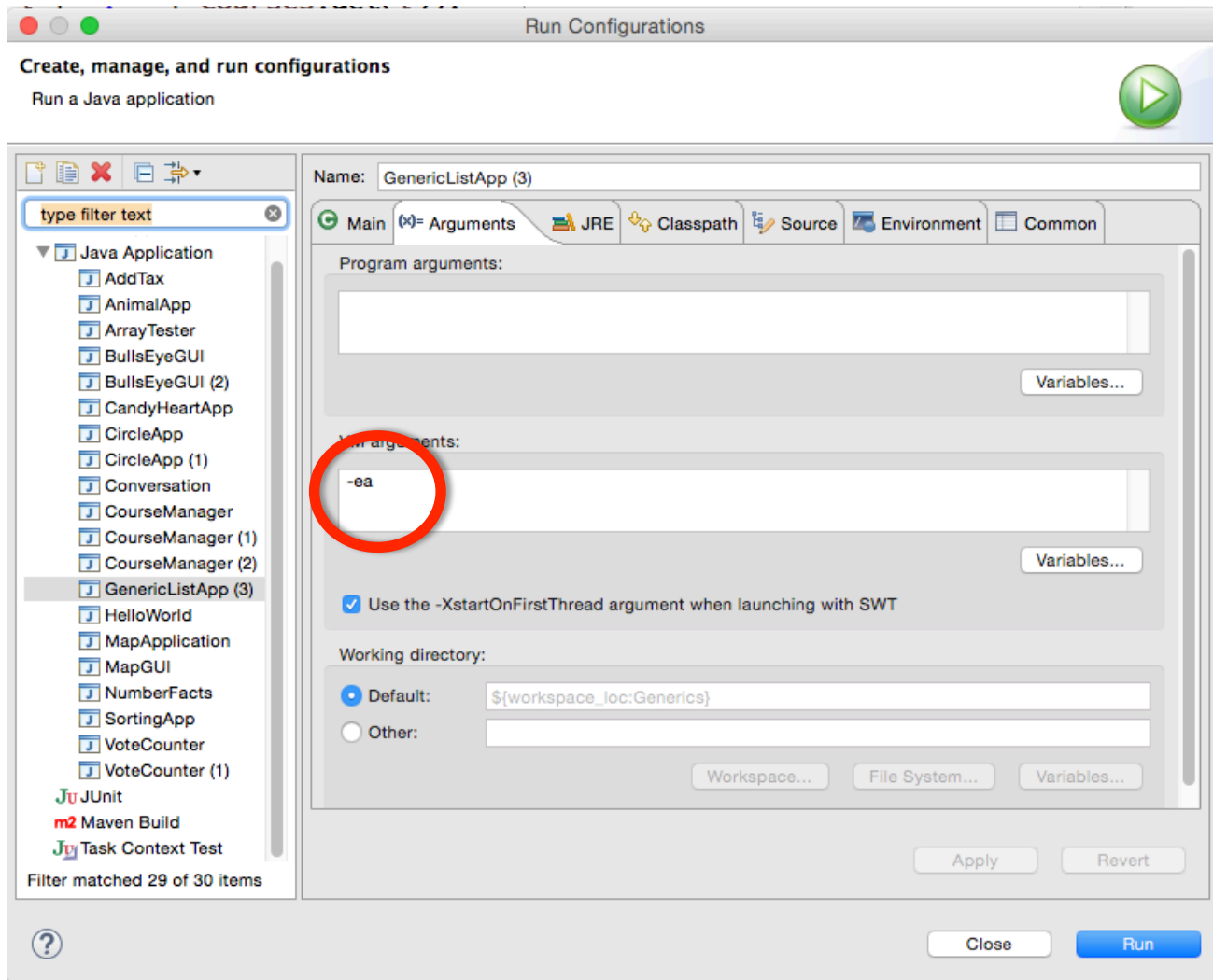
DEMO HOMEWORK 3
DEBRIEF LAB 4
START GENERICS

JAVA ASSERTS

- Just like Python
- Useful for catching bugs at the source
- Example:

```
if (this.head == null) {  
    assert this.counter == 0;  
}
```

HOW TO TURN ASSERTS ON IN JAVA



COPYING A DATA STRUCTURE

What does this code produce?

```
LinkedList<Double> numbers1 = new LinkedList<Double>();  
LinkedList<Double> numbers2 = numbers1;  
numbers1.add(Math.PI);  
  
System.out.println(numbers1);  
System.out.println(numbers2);
```


COPYING A DATA STRUCTURE

What does this code produce?

```
LinkedList<Double> numbers1 = new LinkedList<Double>();  
LinkedList<Double> numbers2 = numbers1;  
numbers1.add(Math.PI);  
  
System.out.println(numbers1);  
System.out.println(numbers2);
```

Output:

```
3.141592653589793  
3.141592653589793
```

COPYING A DATA STRUCTURE

What does this code produce?

```
LinkedList<Double> numbers1 = new LinkedList<Double>();  
LinkedList<Double> numbers2 = numbers1;  
numbers1.add(Math.PI);  
  
System.out.println(numbers1);  
System.out.println(numbers2);
```

Output:

```
3.141592653589793  
3.141592653589793
```

Shallow Copy



COPYING A DATA STRUCTURE

What does this code produce?

```
LinkedList<Double> numbers1 = new LinkedList<Double>();  
LinkedList<Double> numbers2 = new LinkedList<Double>(numbers1);  
numbers1.add(Math.PI);  
numbers2.add(Math.E);  
  
System.out.println("numbers1 " + numbers1);  
System.out.println("numbers2 " + numbers2);
```

COPYING A DATA STRUCTURE

What does this code produce?

```
LinkedList<Double> numbers1 = new LinkedList<Double>();  
LinkedList<Double> numbers2 = new LinkedList<Double>(numbers1);  
numbers1.add(Math.PI);  
numbers2.add(Math.E);  
  
System.out.println("numbers1 " + numbers1);  
System.out.println("numbers2 " + numbers2);
```


Output:

```
numbers1 3.141592653589793  
numbers2 2.718281828459045
```

COPYING A DATA STRUCTURE

What does this code produce?

```
LinkedList<Double> numbers1 = new LinkedList<Double>();  
LinkedList<Double> numbers2 = new LinkedList<Double>(numbers1);  
numbers1.add(Math.PI);  
numbers2.add(Math.E);  
  
System.out.println("numbers1 " + numbers1);  
System.out.println("numbers2 " + numbers2);
```



Output:

```
numbers1 3.141592653589793  
numbers2 2.718281828459045
```

Deep Copy via copy constructor