

# 3/3 TEST TOPIC SHEET

Nick Wald

- 3/3 TEST TOPIC SHEET
  - CLASS STRUCTURE
    - CLASS EXTENSION
  - 2D ARRAYS
  - ARRAYLIST
    - Key Methods:
      - ADD
      - SET
      - GET
      - REMOVE
  - SORTING + SEARCHING
  - STATIC
  - RECURSION
    - INHERITENCE + POLYMORPHISM

## CLASS STRUCTURE

## CLASS EXTENSION

## 2D ARRAYS

## ARRAYLIST

A dynamically sized array which has methods attached to it by default

To Create A New arrayList:

```
ArrayList <String> arrList = new ArrayList <String>(100);  
//parentheses can often be ignored, default size is 10
```

### Key Methods:

#### ADD

Adds value to end of arrayList, or at specified position which shifts all subsequent indices up by 1

```
listName.add(value);  
//adds parameter to end of arrayList  
listName.add(index, value);  
//adds value at specified index, shifts all other indexes up by 1
```

```
arrList.add("hi");
```

```
//final value in arrList is now hi
```

```
arrList.add(0, "nick");
//value at position 0 becomes nick
//value @ pos 0 shifts to pos 1
//value @ pos 1 shifts to pos 2, etc...
```

## SET

Replaces value at specified position with new specified value

```
listName.set(index, value)
//whatever is in position index has now been changed to the new value
```

```
arrList.set(0, "owen")
//the value at position 0 becomes owen
```

## GET

Return value at specified position

```
arrList.get(0);
//returns value at position 0
```

## REMOVE

Removes value at specified index, shifts all subsequent indexes down by 1

```
arrList.remove(1);
```

## ##SORTING + SEARCHING

## ##STATIC

## ##RECURSION Using a function inside itself, repeatedly shortening and checking through input ESSENTIALLY REDUCING PROBLEM SIZE UNTIL BASE CASE IS REACHED

**Example 1: Reversing a string**  $f(\text{String } s) = \begin{cases} s & \text{if } s.length \leq 1 \text{ (base case)} \\ f(\text{String } s.substring(1)) + \text{String } s.charAt(0) & \text{otherwise (recursive case)} \end{cases}$

```
public String f(String s){
    if(s.length() <= 1) return s;
    return f(s.substring(1)) + s.charAt(0);
}
```

**Example 2:**

```
/*
 * @param if i have no digits, return 0
 */
public int addDigits(int num){
    if(num==0) return 0;
    return num%10 + addDigits(num/10); //using method on a smaller scale
    //with integer division, when you divide by 10, last number is chopped off, leaving rightmost digits to check. then run again at a
    //smaller scale w the last digit chopped off
}
```

Start by writing base case, then recursive cases Cases in which the function is called inside itself are recursive cases

**Recursively tracing the number 7315 using addDigits function:**

```
addDigits(7315)
= 5 + addDigits(731)
= 5 + (1 + addDigits(73))
= 5 + (1 + (3 + addDigits(7)))
= 5 + (1 + (3+ (7 + addDigits(0))))
= (5 + (1 + (3 + (7 + 0))))
= 16
```

---

### ###INHERITENCE + POLYMORPHISM