

# CS 113 – Computer Science I

## Lecture 09 – Recursion, Strings, Arrays

Tuesday 02/20/2024

# Announcements

- HW03 released

# Agenda

Recursion review

Arrays review

# Recursion Example - Factorial

- What is a factorial?  $n!$
- product of all integers less than or equal to  $n$ 
  - $n! = n * n-1 * n-2 \dots 1$
  - $5! = 5 * 4 * 3 * 2 * 1$
  - $4! = 4 * 3 * 2 * 1$
  - $3! = 3 * 2 * 1$
- What is the base case?

# Recursion Example – IndexOf letter

Your turn! Write a method called IndexOf.

Arguments: String (haystack), Character (needle)

Return: the index of the character in the String. You can assume needle is in haystack.

# Arrays

# Arrays

Filing Cabinet

Idea: Store multiple values into a single variable

Values are sequential

Analogous to a list

# Arrays

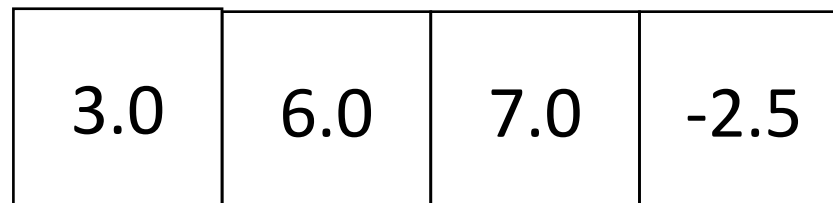
```
double val = 3.0;
```

val



```
double[] vals = {3.0, 6.0, 7.0, -2.5};
```

vals





# Array Indexing

Access individual elements of an array with indexing

`array[index]`

We use *zero*-based indexing

first element is **0**

last element is **length-1**

Accessing indices out of range results in a **runtime error!**

# Arrays

Three ways to initialize an array

1. With an initial value

```
int[] numbers = {1, 2, 5};
```

2. With allocated space, but uninitialized

```
int[] numbers = new int[3];
```

3. With an empty array reference

```
int[] numbers = null;
```

# Arrays - default values

```
int[] numbers = new int[3];
```

numbers

0	0	0
---	---	---

```
String[] words = new String[3];
```

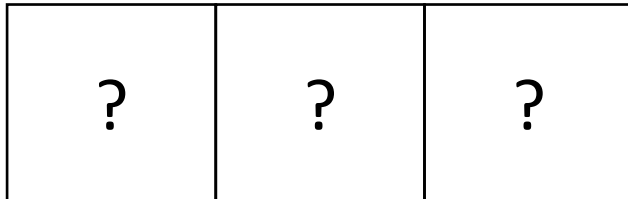
words

null	null	null
------	------	------

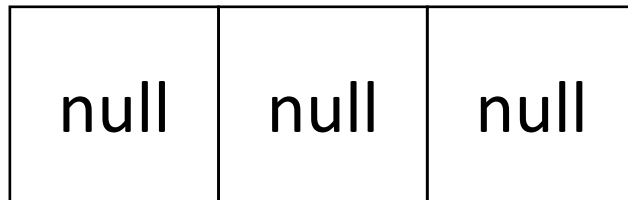
# Arrays - default values

```
Scanner[] words = new Scanner[3];
```

Scanner



Scanner



# Arrays

```
int[] x = {2, 7, 5};
```

```
System.out.println(x.length); //what will this print?
```

`.length` field tells us how many elements are in the array

Once an array is full, you cannot add more elements to it!

# Exercise: Even Indices

## Your turn!

Write a program, `EvenIndices.java`, that asks the user for 5 integers and then prints *every other* number

# Strings

# Strings - useful methods

`toLowerCase ()`

`toUpperCase ()`

<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#toLowerCase-->



# Strings

Strings are implemented as *arrays of characters*

**char:** New built-in type, denoted with single quote, e.g. 'a' or '{'

Get the length of a string with `length()`

```
String greeting = "hola";
```

```
int len = greeting.length(); // what is the length?
```

```
char c = greeting[2]; // what character is in index 2?
```

# Strings as an array of characters

**Java does not let us deal with a String by accessing the underlying array**

String str = "hello world"

- How many characters in this String?  
10
- How do we access the first character?  
str.charAt(0)
- How do we access the 5<sup>th</sup> character?  
str.charAt(4)

# Exercise: GetCharacters.java

Write a program, GetCharacters.java, that asks the user for a word and then prints the first, last and middle character.

```
Enter a word: hola!  
FirstIndex: 0 FirstCharacter: h  
MiddleIndex: 2 MiddleCharacter: l  
LastIndex: 5 LastCharacter: !
```

# Command line arguments

```
public static void main(String[] args)
```

Command line arguments are an *array of String*

Exercise: Write a program called `commandLineArgs.java` that

- 1) prints out 3 command line arguments that are passed in.
- 2) Compute the sum of three command line arguments (assuming they are integers)

**Your turn!**

# Recursion Example – printList

Write a recursive function that prints the contents of an array