

CS 113 – Computer Science I

Lecture 09 – Arrays & Recursion

Tuesday 10/08/2024

Announcements

- HW05
 - Due Monday after fall break
- Office hours:
 - Adam's Tuesday 2:40-4:00pm Thursday 2:40-4:00pm

HW04 feedback

HW02 lessons learned – emphasis on asking questions – great!

HW04

- “the instructions for 'sumOfPreviousN' is extremely confusing...”
- ” I realized that homework is an important part of learning CS”

Ask for help – use Piazza!

Midterm – Thursday 10/24

In class, closed book

Terminal commands, vim, directory structure
variables (int, double, char, bool, string, array)

Expressions

Methods

Frame diagrams

Conditionals

Recursion

Practice exam is on course website

Agenda

Recursion - review

Arrays – reviews

Strings and Arrays as Objects

Arrays

Idea: Store multiple values into a single variable

Values are sequential

Analogous to a list

Arrays

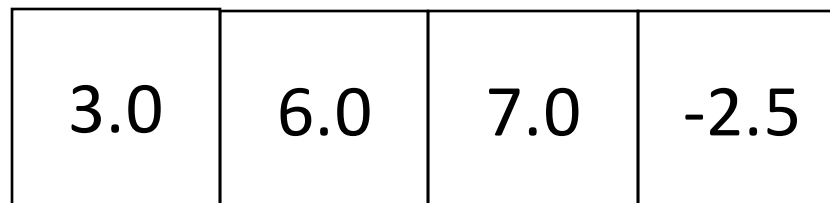
```
double val = 3.0;
```

val



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

vals



Arrays

Three ways to initialize an array

1. With an initial value

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

2. With allocated space, but uninitialized

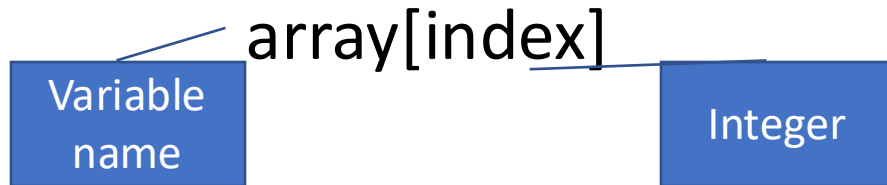
```
int[] numbers = new int[3];
```

3. With an empty array reference

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


Array Indexing

Access individual elements of an array with indexing



We use *zero*-based indexing

first element is **0**

last element is **length-1**

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

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)

Recursion Example – printList

Write a recursive function that prints the contents of an array

Agenda

Recursion - review

Arrays – reviews

Strings and Arrays as Objects

Initializing empty arrays

```
int[] nums = new int[3];  
    [0, 0, 0]
```

```
String[] strs = new String[3];  
    [null, null, null]
```

```
public static void add1(int[] list, int pos) {  
    if (pos >= list.length) {  
        return;  
    }  
    list[pos] += 1;  
    add1(list, pos+1);  
}
```

What is numbs after we call
add1?

```
public static void add1(int[] list) {  
    add1(list, 0);  
}
```

```
public static void main(String[] args) {  
  
    int[] numbs = {10, 20, 30};  
    printList(numbs);  
    add1(numbs);  
    printList(numbs);  
}
```

Objects

Strings and arrays are **NOT** primitives

They are objects

Explains why we can't use "==" to compare Strings

"==" checks if two objects are the same
not if the two values are the same