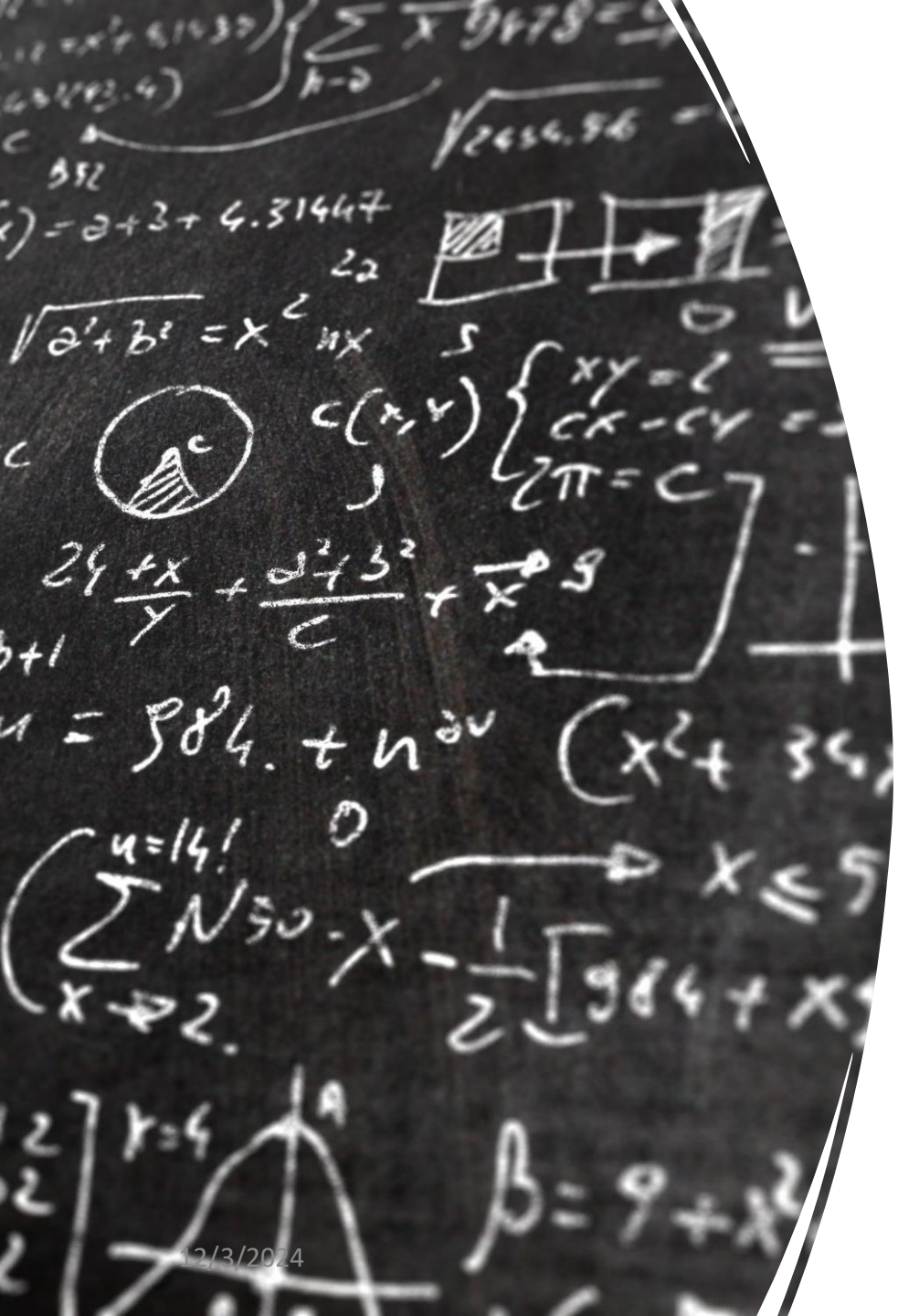

CS 113 – Computer Science I

Lecture 22 – Midterm 2 Review

Tuesday 12/03/2024

Announcements

Mid-semester feedback survey

HW11 – due Thursday 12/12

No lab this week

Additional Office hours today: 3-4pm

Final: Wednesday 12/18 9:30am-12:30pm Park 238

Midterm 2

Thursday 12/05

Material:

- Midterm 1 material
- Loops
- Classes & OOP
- Searching

Midterm 2 study tips

Read the textbook

Vocab section in each textbook chapter:

- Write out the definitions yourself

Do the practice problems in the textbook

- Code solutions: https://github.com/ChrisMayfield/ThinkJavaCode2

Classes vs Objects

Class is a

Classes vs Objects

Class is a blueprint

Classes vs Objects

Class is a blueprint for a custom data type

An ***object*** is an instance of that custom data type

Class defines what type of data is stored in the custom data type and how we can interact with that data

Classes vs Objects

Class is a blueprint for a custom data type

An ***object*** is an instance of that custom data type

Class defines what type of data is stored in the custom data type and how we can interact with that data

Classes vs Objects

Class is a blueprint for a custom data type

An ***object*** is an instance of that custom data type

Class defines what type of data is stored in the custom data type and how we can interact with that data

instance variables

Classes vs Objects

Class is a blueprint for a custom data type

An ***object*** is an instance of that custom data type

Class defines what type of data is stored in the custom data type and how we can interact with that data

instance variables

Classes vs Objects

Class is a blueprint for a custom data type

An ***object*** is an instance of that custom data type

Class defines what type of data is stored in the custom data type and how we can interact with that data

- instance variables
- instance methods

Instance variables

Instance variable is specific to an object

- Different objects of the same class have the same instance variables
- but the values of the instance variables can differ

Example:

- Every car has doors

```
Car sportsCar = new Car("porshe")
Car van = new Car("Sienna")
```

- How many doors do sportscar and van probably have?

    2 and 4 respectively

Static variables

Instance variable is specific to an object

Static variable is specific to a class

- Typically doesn't change

Example:

- What is the largest integer?
  - Lets look at the Integer class documentation: https://docs.oracle.com/javase/8/docs/api/java/lang/Integer.html
- What is the value of PI?
  - Lets look at the Math class documentation: https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html

Static vs instance method

Instance method requires and acts on a specific object

Static methods typically take in the object as a parameter

Classes vs Objects

Class is a blueprint for a custom data type

An ***object*** is an instance of that custom data type

Class defines what type of data is stored in the custom data type and how we can interact with that data

How do we create a new object?

Constructor

A special method defined in a class

When called, it creates a new object and returns that object

Value constructors

- Parameters are placeholders for the values that will be assigned to instance variables

Empty constructors:

- No parameters
- Instance variables are assigned default values
- Default values are chosen by the class designer

Remaining Questions

Could you go over the basics of object oriented programming and how to call methods between different java files?

Could you go over when/why to use static vs non static in method signatures

- Static vs instance variables:
  - A variable declared within a class as static. There is only one copy of a class variable, mo matter how many objects there are.

- constructors and constructor methods
- Could you go ever the definition for class and object and when to use which? Thanks.
- Why is a 2D array displayed row by row instead of in a single row? In other words, how does the computer know to start a new line when iterating through the outer loop?
- Is the length of an empty string 0? What is the length of a null string?

Remaining Questions

Common search problems

Is an item in an array?

- Returns: True or False

Where in an array is the item?

- Returns: the index (an integer)
  - Standard: -1 if the item is not found

How many times does the item appear in an array?

- Returns: a count (an integer)

What is the min, max, or average value in an array?

- Returns: the value (double)

Linear Search

Check each item in a collection one by one

Why is this call linear search?

Time it takes to search increases *linearly* with the size of the list

Linear Search

What happens (in terms of speed) when the list is very large?

The search becomes slower

In what cases do we do the most work (i.e. perform the most comparisons)?

When the item is not in the list

In what cases do we do the least amount of work?

When the item is the first element in the list

Binary Search

If we could change the list, is there a way to search more efficiently?

Yes, if the list is sorted

Guessing game – in class exercise

Pair up:

- Person A chooses a number between 1 and 100
- Person B guesses the number
- Until the guess is correct:
  - Person A tells whether the guess is too high or too low
  - Person B guesses again

Binary Search

Assuming list is sorted in ascending order

High-level Algorithm:

- Step 1: Find the midpoint of the list:
  - if the search value is at the midpoint – we are done!
  - if the value we are searching for is above the midpoint,
    - Search right: cut our list in half and repeat step 1 with the right half of the list
  - If the value we are searching for is below the midpoint
    - Search left: cut out list in half and repeat step 1 with the left half of the list

Binary Search – Initial Values

lowIndex, highIndex, midIndex

lowIndex = 0

highIndex = length of the array – 1

midIndex = $\frac{lowIndex+highIndex}{2}$

Binary Search – Initial Values

lowIndex, highIndex, midIndex

If value at midIndex== searchValue:
    Success!

If value at midIndex < searchValue:
    lowIndex = midIndex + 1
    update midIndex

If value at midIndex > searchValue:
    highIndex = midIndex – 1
    update midIndex

Binary search

0 1 2 3 4 5

String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 99

low	mid	high	ls[mid]

Binary search

0 1 2 3 4 5

String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 99

low	mid	high	ls[mid]
0	2	5	44

Binary search

0 1 2 3 4 5

String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 99

low	mid	high	ls[mid]
0	2	5	44
3	4	5	99 (found!)

Binary search

0 1 2 3 4 5
String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 30

low	mid	high	ls[mid]

Binary search

0 1 2 3 4 5
String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 30

low	mid	high	ls[mid]
0	2	5	44

Binary search

0 1 2 3 4 5
String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 30

low	mid	high	ls[mid]
0	2	5	44
0	0	1	-20

Binary search

0 1 2 3 4 5
String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 30

low	mid	high	ls[mid]
0	2	5	44
0	0	1	-20
1	1	1	-4

Binary search

0 1 2 3 4 5
String[] ls = {-20, -4, 44, 58, 99, 145}

Search for 30

low	mid	high	ls[mid]
0	2	5	44
0	0	1	-20
1	1	1	-4
2		1	Not found!

Binary search w/ Strings

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “cow”

low	mid	high	ls[mid]

Binary search w/ Strings

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “cow”

low	mid	high	ls[mid]
0	3	7	“cat”

Binary search w/ Strings

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “cow”

low	mid	high	ls[mid]
0	3	7	“cat”
4	5	7	“dog”

Binary search w/ Strings

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “cow”

low	mid	high	ls[mid]
0	3	7	“cat”
4	5	7	“dog”
4	4	4	“cow”!

Binary search

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “elephant”

low	mid	high	ls[mid]

Binary search

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “elephant”

low	mid	high	ls[mid]
0	3	7	“cat”

Binary search

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “elephant”

low	mid	high	ls[mid]
0	3	7	“cat”
4	5	7	“dog”

Binary search

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “elephant”

low	mid	high	ls[mid]
0	3	7	“cat”
4	5	7	“dog”
6	6	7	“fish”

Binary search

0 1 2 3 4 5 6 7

String[] ls = {"bear", "bird", "bug", "cat", "cow", "dog", "fish, "lion“};

Search for “elephant”

low	mid	high	ls[mid]
0	3	7	“cat”
4	5	7	“dog”
6	6	7	“fish”
6		6	