



CS 113 – Computer Science I

Lecture 26 – Final Exam Review

4/30/2024

Announcements

Lab 9, Lab 10, Lab 11, HW9 due May 7th

OH this week: Wednesday 2:30-3:30pm
Friday 10-11am, 1-3pm

Exam Format

- Cumulative
- 180min
- 2 8.5/11in cheat sheets allowed (front and back)
- Format: 125 total points
 - 20 points T/F questions
 - 34 points short answer
 - 6 points reading and understanding code
 - 65 points programming

Topics

- Errors (syntax, runtime, logic)
- Types (primitives and objects)
- Variables
- Methods (declarations and calls)
- Strings
 - substring, length, equality
- Truth tables
- If statements / conditionals
- Arrays and arrays of arrays
- Recursion
- Loops and nested loops

Topics

- Mutability
- Classes
 - a. constructors, accessors, modifiers, instance variables, this keyword
- equals
- toString
- Statics vs non-static
- OOP
 - a. super keyword
 - b. Access modifiers
 - c. Inheritance and polymorphism
- try-catch
- interfaces

Topics

- Searching
 - Linear Search
 - Binary Search
- Sorting
 - Bubble Sort
 - Selection Sort
- Runtime Complexity and Big-O notation

Compilation and Execution

What is Compilation?

Converts java file (.java) to a file that the computer understands

What command do we use to compile?

`javac`

What kinds of errors are caught at compile time?

syntax errors, type errors

Compilation and Execution

What is Execution?

read and execute the class file

What command do we use to execute?

java

What kinds of errors are caught at run time?

exceptions

Errors

compilation, runtime, or logic?

```
Example.java:3: error: incompatible types:  
String cannot be converted to int
```

```
int x = 100 + "s";
```

Errors

compilation, runtime, or logic?

```
Exception in thread "main"  
java.lang.ArithmeticException: / by zero  
    at Example.main(Example.java:3)
```

Errors

compilation, runtime, or logic?

```
Example.java:3: error: cannot find symbol
```

```
    System.ouy.println("Hello, World!");
```

```
        ^
```

```
symbol:   variable ouy
```

```
location: class System
```

Errors

compilation, runtime, or logic?

```
Exception in thread "main"  
java.util.IllegalFormatException: d !=  
java.lang.String
```

Errors

what's a logic error?

Compile err, runtime err, or none?

```
int[] numbers = new int[3];  
  
System.out.println(numbers[3]);
```

Runtime error! AOB

Compile err, runtime err, or none?

```
String str = null;  
int length = str.length();  
System.out.println(length);
```

Runtime error! NPE

Compile err, runtime err, or none?

```
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a number: ");
try {
    int num = scanner.nextInt();
} catch (Exception e) {
    System.out.println("err");
}
```

None

Compile err, runtime err, or none?

```
try {
    Scanner scanner = new Scanner(new File("input.txt"));
} catch (Exception e) {
    System.out.println("err");
}
int num = scanner.nextInt();
```

Compiler error - scanner is not in scope!

Compile err, runtime err, or none?

```
Scanner scanner = new Scanner(System.in);  
System.out.print("Enter a number: ");  
int num = scanner.nextInt();
```

Potential runtime error: `NumberFormatException`

Compile err, runtime err, or none?

```
int foo(int n) {  
    if (n == 0) {  
        return 0;  
    }  
    return n % 10 + foo(n * 10);  
}
```

Potential runtime error: StackOverflowException

Compile err, runtime err, or none?

```
Scanner sc;  
try {  
    sc = new Scanner(new File("input.txt"));  
    int num = sc.nextInt();  
} catch (FileNotFoundException e) {  
    System.out.println("err");  
}
```

Potential runtime error: NumberFormatException

Compile err, runtime err, or none?

```
int[] numbers = {1, 2, 3, 4, 5};  
for (int i = 0; i <= numbers.length; i++) {  
    System.out.println(numbers[i]);  
}
```

runtime error: AOB

Compile err, runtime err, or none?

```
int[] numbers = {1, 2, 3, 4, 5};  
for (int i = 1; i < numbers.length; i++) {  
    System.out.println(numbers[i] / numbers[i - 1]);  
}
```

None

Compile err, runtime err, or none?

```
int[] numbers = {1, 2, 3, 4, 0};  
for (int i = 1; i < numbers.length; i++) {  
    System.out.println(numbers[i] / numbers[i - 1]);  
}
```

None

Variables

1. How do I declare a variable?
2. How do i assign a value to a variable?

Variables

assignment or declaration?

- `int x;`
- `x = foo();`

Types

1. List some types
2. What is a primitive type?
 - a. list some
3. What is an object (reference type?)
 - a. list some
4. What are the differences?

Object equality

What does “==” do when comparing objects?

checks if the addresses of two objects are the same... not if the two values are the same

What method should we define on classes to do a proper equality check?

`boolean equals(Object o)`

Types

types.java

Types - what will this print?

```
double y = 10;  
double z = 1.25;  
int x = y * z;  
  
System.out.println(x);
```

Types

What are the default values of the following types?

1. `int`
 - a. `0`
2. `boolean`
 - a. `false`
3. `String`
 - a. `null`
4. `Scanner`
 - a. `null`
5. `int[]`
 - a. `null`
6. `String[] s = new String[2];`
 - a. `[null, null]`
7. `int[] i = new int[2];`
 - a. `[0,0]`

Scope

Scope.java

Scope

Scope2.java

Conditionals

Conditional.java

Truth tables - true, false, or err?

```
boolean A(int x, int y) {  
    return (x > 5 && y < 10)  
        || (x < 0 && y >= 20);  
}  
  
boolean B(boolean p, boolean q) {  
    return !p || (p && q);  
}
```

```
int a = 7;  
int b = 12;  
boolean p = true;  
boolean q = false;
```

1. (a > 5) && (b < 10)
false
2. A(a, b)
false
3. A(b, a)
true
4. B(p, q)
false
5. B(q, p)
true

Strings

- concatenation
- length
- substring

Strings.java

reading code

What will this print?

```
public static void mysteryMethod(int n) {
    if (n > 0) {
        mysteryMethod(n - 1);
        System.out.print(n + " ");
        mysteryMethod(n - 1);
    }
}

public static void main(String[] args) {
    mysteryMethod(5);
}
```

OOP Practices - super and equals method

You are tasked with developing a program to manage drinks in a coffee shop. Your task is to design and implement classes to represent different types of drinks using inheritance in Java, and override the `equals` method for comparison.

1. Drink
 - a. String temperature (hot / cold), int price, String size (S, M, L)

2. Coffee
 - a. All of the vars on Drink plus roast (light, medium, or dark)

3. Tea
 - a. String flavor

Reading from a file

Suppose you have a text file named "data.txt" containing the following content:

10

20

30

40

50

Write a Java program that reads the numbers from the file "data.txt" and calculates their sum. Assume that each number is on a separate line in the file.

Searching, Sorting, and Runtime Complexity

Runtime Analysis: Big O Notation

- Mathematical notation used to describe the performance or complexity of an algorithm.
- Hardware independent
- Represents the upper bound of the time complexity in the **worst-case scenario**.
- Helps us understand how the runtime of an algorithm grows ***as the input size increases***.

Runtime Complexity

Sort these from fastest to slowest:

- $O(n)$
- $O(n^2)$
- $O(\log n)$
- $O(1)$
- $O(2^n)$

Searching

- Linear Search
 - Best case?
 - Worst case?

- Binary Search
 - Best case?
 - Worst case?

Searching

[5, 10, 17, 22, 26, 40, 50, 100]

1. Perform a **linear search** for the element 50
 - a. How many elements did we check?

2. Perform a **binary search** for the element 50 and show each step
 - a. how many elements did we check?

Searching

[5, 10, 17, 22, 26, 40, 50, 100]

1. Perform a **linear search** for the element 5
 - a. How many elements did we check?

2. Perform a **binary search** for the element 5 and show each step
 - a. how many elements did we check?

Searching

Is binary search *always* faster than linear search?

No! Big-O notation is an analysis of the worst case.

In some cases, a linear search will be faster.

Sorting

Show each step of sorting the following list:

[12, 35, 78, 21, 93, 73, 8, 66]

1. Selection Sort
2. Bubble Sort

Sorting

- Selection Sort
 - runtime complexity?
- Bubble Sort
 - runtime complexity?

Big-O Example 1

```
int n = Integer.parseInt(args[0]);
int power = 1;
while (power < n) {
    System.out.print(power + " ");
    power *= 2;
}
```

How does the runtime grow as a function of the input size?

$O(\log n)$

Big-O Example 2

```
int fetchFirstElement(int[] arr) {  
    return arr[0];  
}
```

How does the runtime grow as a function of the size of arr?

$O(1)$

Big-O Example 3

```
int n = Integer.parseInt(args[0]);
int tot = 0;
int i = 0;

while (i < n) {
    tot = tot * i;
    i++;

    for (int j=0; j<10000; j++) {
        System.out.println("hello");
    }
}
```

How does the runtime grow as a function of the input size?

Linearly!

$O(n)$

Big-O Example 4

```
int n = Integer.parseInt(args[0]);

for (int i = 0; i > (-1*n); i--) {
    for (int j = 0; j < n; j++) {
        System.out.println(i, j);
    }
}
```

How does the runtime grow as a function of the input size?

Quadratically!

$O(n^2)$

We do n operations n times

Big-O Example 5

```
String[] lst =
    {"19", "12", "20", "15"};

for (int i=0; i<100; i++) {
    System.out.println(getNum(lst));
}

int getNum(int[] arr) {
    return Integer.parseInt(arr[0]);
}
```

How does the runtime grow as a function of the size of `lst`?

Constant! The runtime is not affected by the number of elements in `lst`

$O(1)$

Big-O Example 6

```
int[] lst = {1,2,3,4,5,6,7};

for (int i=0; i<lst.length; i++) {
    findMax(lst);
}

int findMax(int[] arr) {
    int max = Integer.MIN_VALUE;
    for (int i=0; i<arr.length; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }
    return max;
}
```

How does the runtime grow as a function of the size of lst?

$O(n^2)$

Programming Questions

Q1 - Problem Solving, Recursion, and Loops

Write a function called "numOccurs(int[] a, int[] b)". The function should determine how many times the elements of a occur in b. You can assume that both arrays will not be empty.

1. Write it recursively
2. Write it with a loop

Q2 - Classes, OOP, Arrays of Objects

Testing you on:

1. How to initialize an array as an instance variable
 - a. What size should I make it?
2. How to deal with dynamically sized arrays
 - a. What if its full when I try to add to it?
3. Make sure to avoid NPEs
4. How and when to use inheritance

Q2 - Classes, OOP, Arrays of Objects

Design and Implement a class that represents a Team. The team should have Players each with a name. Players can either be Offense, Defense, or Coaches. The Team class should support the following operations:

1. add: takes a player and adds them to the team
 - a. There is a max capacity of 2 players of each position (offense, defense)
 - b. Only one coach is allowed
2. trade: remove the player from the team
3. getOffense: returns a list of offensive players
4. getCoach: returns the coach's name

Q2 - Classes, OOP, Arrays of Objects

Testing you on:

1. How to initialize an array as an instance variable
 - a. What size should I make it?
2. How to deal with dynamically sized arrays
 - a. What if its full when I try to add to it?
3. Make sure to avoid NPEs
4. How and when to use inheritance

Q3 - Problem Solving, Arrays of Arrays

Write a function called "getPerim(int[][] a)". The function should return the an int[] of the perimeter values of a.

1	7
35	1



[1,7,35,1]

1	7	4
3	15	2
9	-1	6
35	1	3



[1,7,4,3,2,9,6,35,1,3]

Q4 -Problem Solving, Runtime Complexity, Loops

Write a method called `uniqueElements()` that takes in an array of integers and returns the number of unique elements from the original array.

To receive full credit, your solution's runtime must be $O(n)$ Partial credit will be given for less efficient solutions. **You may use additional data structures if needed.**

Q5 - dynamic array size

Write a method called `maxBoard()` that takes in a filename and generates a 2D array filled with 'O's.

The file contents will contain two row. The size of the 2D array should be $[x][y]$ where x is the max value in the first row and y is the max value in the second row