CS 0007 RECITATION – 9/9/21

LIN ROJTAS – 10:00A – 10:50A

AGENDA

- Introductions (again)
- Review of the command line
- Review of Java API
- Variables and arithmetic

ABOUT ME!

- My name is Lindsey, but you can call me Lin ☺
- Any pronouns, I always like to say “whatever makes a joke funnier”
- Junior – CS Major, Linguistics Minor
- Born and raised 30 minutes south of Pittsburgh (Thomas Jefferson High School, WJHSD)
- I like rhythm games, stuffed animals, cooking shows, and my fur babies

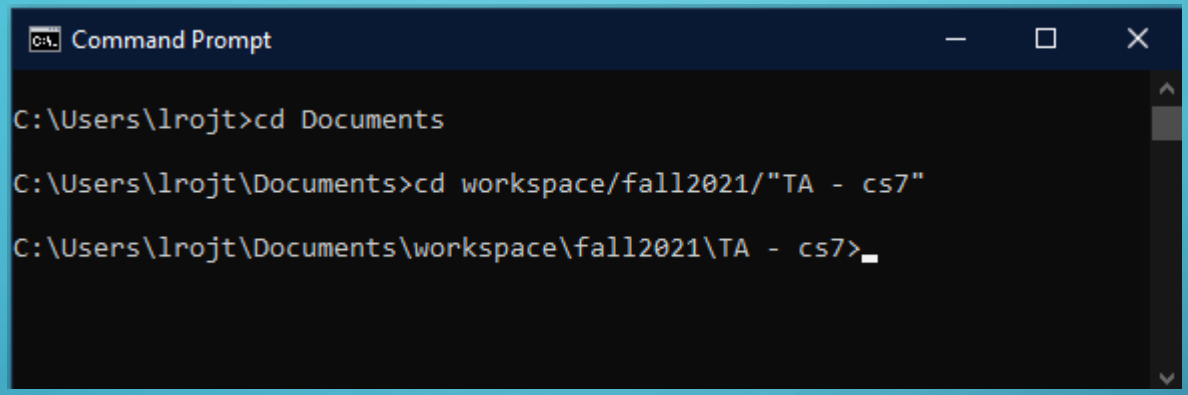
ABOUT THIS RECITATION

- I'm the one that determines grades and due dates for the labs, so don't reach out to Paulo about them without talking to me about it first
- Recitations will be held here, 10a-10:50a
  - Will likely let out early most weeks
- Office hours
  - 2832 Cathedral of Learning, Mon 6:00p-8:00p, Wed 7:00p-8:30p
  - Also by appointment, both virtually and in person

POLICIES

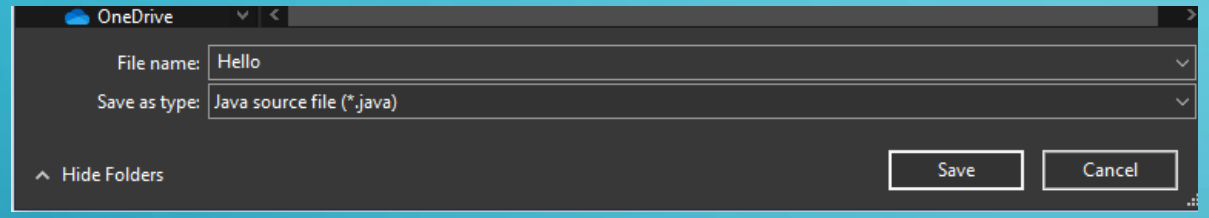
- Attendance is not mandatory, but is strongly recommended as I will go over concepts from the class and hints for the labs
- Labs will be due every Wednesday at 11:59p
  - I'm not picky about specific solutions – if it works, it works!
  - Cheating is for losers – don't do it
  - If you have an issue with this or need an extension, let me know!
  - However, if everyone turns in their labs on time (barring DRS accommodations and major real-life events) for the whole term, I might buy you guys donuts…
- Slides will be posted on my Github
  - https://github.com/rojtas/cs7-fall2021-recs

COMMAND LINE REVIEW



- USEFUL COMMAND: `cd (folder name)` – go into a folder in your current directory
  - Use `cd ..` to go to a folder *outside* the current directory (ex. Say I want to go to **`fall2021`**)
  - Try not to use spaces in your folder names, but if you do, use quotes in the command prompt!

COMMAND LINE REVIEW



Hello.java

```
1  public class Hello {
2      public static void main(String[] args) {
3          System.out.println("Hello world!");
4      }
5  }
```

- Before we run our program, we need to make sure that:
  - We are saving as a .java file
  - The name of our program is the same as the word that follows `public class` in that program.

COMMAND LINE REVIEW

```
C:\Users\lrojt\Documents\workspace\fall2021\TAcs7>javac Hello.java

C:\Users\lrojt\Documents\workspace\fall2021\TAcs7>java Hello
Hello world!
```

- USEFUL COMMAND: `javac (file name).java` – compiles our written code into bytecode
- USEFUL COMMAND: `java (file name)` – runs the machine code that was compiled
  - ALWAYS `javac` BEFORE YOU `java`!!!
  - When using `javac`, make sure you include `.java` at the end of your file name!

JAVA API – MATH EXAMPLES

- With APIs, Google is your friend!!
  - https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html
- Highlights
  - `Math.sqrt(double a)` – returns the square root of a number `a`.
    - `Math.sqrt(4)` returns `2.0`
  - `Math.pow(double a, double b)` – returns the value of a number `a` raised to the power of another number `b` (or a^b)
    - `Math.sqrt(3, 2)` returns `9.0`
- Feel free to explore and test on your own!

JAVA API – SCANNER

- The Scanner API is typically used for accepting user input
  - Asking for two numbers to be added together, entering your first name, etc.
- Unlike the Math class, you need to import this class into your program by including `import java.util.Scanner;` at the very top of your program (above your class!)

```
import java.util.Scanner;

public class Hello {
```

- Note: you can import all the classes in `java.util` with `import java.util.*;`

PARTS OF A PROGRAM

1 2 3

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello world!");
        System.out.println("This is a program.");
    }
}
```

- 1 – Class
- 2 – Method
- 3 – method delimiter

- Comments can be formatted

```
// like this
/ * like this */
/* and
 * like
 * this */
```

DATA TYPES

- Primitives vs. objects
  - Primitive examples: boolean, byte, short, long, int, double, float, char (all lowercase!)
    - boolean: true/false value
    - char: character (‘a’.,‘x’)
    - short, long, int, double, float, and byte are all numbers of varying lengths
      - Most of the time, you’ll use int and double of these six
      - Only double and float can be decimal values
  - Object examples: String, Scanner, and many… many more (capital first letter!)

VARIABLE NAMING CONVENTIONS

- It's important to name your variables in ways that both you and anyone else that may see your programs (Paulo, myself, the grader) will understand.
  - DO NOT use single letters or non-descriptive names!!!
  - DO NOT use Java built-in keywords!!! (List: https://en.wikipedia.org/wiki/List_of_Java_keywords)
- Variable names are case-sensitive (`myVariable` is not the same as `myvariable`)
- Variables cannot start with numbers or special symbols (except for _ and $).
- In this class (and in any other coding classes unless you are told otherwise), avoid the use of anything non-alphanumeric.

VARIABLE NAMING CONVENTIONS

- If you're naming your variable one word, you'll typically name your variable that word in all lowercase
  - **Examples:** `height`, `movie`, `speed`
- If your variable name is more than one word, the first word will be lowercase with the subsequent words' first letter capitalized
  - **Examples:** `myHeight`, `username`, `correctAnswer`, `myFavoriteClass`

VARIABLE NAMING CONVENTIONS

```
int myNumber = 7;
double myDecimal = 0.007;
String myString = "Hello";
```

- Format: (variable type) (variable name) = (variable value);
- `System.out.println(myNumber);` will print `7`

OPERATIONS AND OPERATOR PRECEDENCE

Operator	Purpose	Example	Equivalent
+=	Addition	x += 2	x = x + 2
-=	Subtraction	x -= 2	x = x - 2
/=	Division	x /= 2	x = x / 2
*=	Multiplication	x *= 2	x = x * 2
%=	Modulus	x %= 2	x = x % 2

- There are shorthand ways of using these operations!

OPERATIONS AND OPERATOR PRECEDENCE

- You don't need to know all of these!!
- The most important ones are additive (+, -) and multiplicative (*, /, %)
- You may end up using some of the other ones in the future, but… we'll cross that bridge when we get there ☺

Level	Operator	Description	Associativity

16	`[]`
	`.`
	`()`
		access array element
		access object member
		parentheses
			left to right

15	`++` `--`	unary post-increment unary post-decrement	not associative

14	`++`
	`--`
	`+`
	`-`
	`!`
	`~`
		unary pre-increment
		unary pre-decrement
		unary plus
		unary minus
		unary logical NOT
		unary bitwise NOT
			right to left

13	`()` `new`	cast object creation	right to left
12	`* / %`	multiplicative	left to right
11	`+ -` `+`	additive string concatenation	left to right
10	`<< >>` `>>>`	shift	left to right

9	`< <=`
	`> >=`
	`instanceof`
		relational	not associative

8	`==` `!=`	equality	left to right
7	`&`	bitwise AND	left to right
6	`^`	bitwise XOR	left to right
5	`|`	bitwise OR	left to right
4	`&&`	logical AND	left to right
3	`||`	logical OR	left to right
2	`?:`	ternary	right to left

FOR NEXT WEEK

- Lab 1 is out…!
  - It's a really easy one. All you have to do is show me that you have Java installed and can print "Hello world." Due next Wednesday at 11:59!
  - Keep an eye on Canvas; it'll be posted some time tonight (email me if Friday comes and I forgot to post the assignment).
- Next week: casting, the final keyword, strings and input, and style!
- Wear a mask, wash your hands, get vaccinated, and be safe!