CSC 212
PROGRAMMING WITH DATA STRUCTURES

SPRING 2016

PROF. SARA SHEEHAN

SMITH COLLEGE

CLASS 2: JAN 28 OUTLINE

- Syllabus
- Java to Python reverse engineering
- Primitives in Java
- First data structure
- Java style

SYLLABUS

COURSE WORK

- **Labs**
  - During 2 hour lab section
  - Ideally you will finish, if not, due midnight on Thursday
  - Coarsely graded

- **Homeworks**
  - Weekly (unless there is a reason to postpone)
  - Due on Wednesdays at midnight (generally)
  - Graded on completeness/correctness, style, clarity

- **Participation**
  - In class and in labs (at least one question or answer)
  - Piazza (at least one post required)
  - Office hours (at least one visit required)

[50% of grade]

COURSE WORK

- **Midterm**
  - During the lab section before spring break
  - One note sheet (otherwise closed book/technology)
- **Final Exam**
  - Self-scheduled

[Exams: 30% of grade]

- **Final Project**
  - Synthesize material
  - Due before finals

[Project: 20% of grade]

TEXTBOOKS

"**Data Structures and Algorithms in Java**”
by Adam Drozdek (2nd edition free online)

"**Murach's Java Programming**” (4th edition recommended)
by Joel Murach, Amazon, some free material online

- Recommended but not required
- On reserve in the library
- Could share with a classmate

PIAZZA

- Class discussion
- Homework help
- Clarifications
- Announcements

Please do not email me or private post on Piazza, unless it’s a question no one else could have (in that case private post is better than email).

https://piazza.com/smith/spring2016/csc212/home

TOPIC LIST (TENTATIVE)

- Python to Java
- Classes in Java
- Primitive Types
- Arrays
- Interfaces, Inheritance
- Abstract Classes
- GUIs
- Linked Lists
- Iterators
- Sorting Algorithms
- Stacks
- Queues
- Complexity and Big O Notation
- Recursion
- Trees
- Decision Trees
- Priority Queues
- Heaps
- Hash Tables
- Graphs

TEACHING ASSISTANTS

- **Office hours 4 nights per week**
- **Last semester: Sun, Mon, Wed, Thurs**
- **7:30-9:30pm**
- **342 Ford (this room)**
- **Lab assistant (Alice Yang)**

Sharon will be here tonight!

http://cs.smith.edu/classwiki/index.php/Computer_Science_TA_hours

COLLABORATION AND HONOR CODE

- **Homeworks:**
  - No one should see your code besides you and the instructors
  - Discuss aspects of the assignment without code
  - Pair programming assignments are an exception

- **Labs:**
  - Encouraged to help each other, including viewing code
  - You should never be using someone else’s keyboard or mouse
  - No one should be using your keyboard or mouse

ELECTRONIC DEVICES

- Okay in class or labs if directly related to the course
- Mostly for others!

ADDITIONAL RESOURCES

* **Disability Services**:

http://www.smith.edu/ods/

(please give me accommodations letters early)

* **Spinelli Center for Quantitative Learning**:

http://www.smith.edu/qlc/

GROUP ACTIVITY: REVERSE ENGINEERING

In groups of 2-3, write down what this same program would look like in Python.

JAVA STYLE CONVENTIONS

- **CamelCaseForEverything!**
  - Classes should start with a capital and methods with lowecase
  - Example: `AddTax`
  - `addTax` (Java) vs. `add_tax` (Python)

https://whathecode.wordpress.com/2011/02/10/camelcase-vs-underscores-scientific-showdown/

JAVA STYLE CONVENTIONS

- **CamelCaseForEverything!**
  - Classes should start with a capital and methods with lowecase
  - Example: `AddTax`
  - `addTax` (Java) vs. `add_tax` (Python)

https://whathecode.wordpress.com/2011/02/10/camelcase-vs-underscores-scientific-showdown/

- **80 characters per line**

JAVA STYLE CONVENTIONS

- **CamelCaseForEverything!**
  - Classes should start with a capital and methods with lowecase
  - Example: `AddTax`
  - `addTax` (Java) vs. `add_tax` (Python)

https://whathecode.wordpress.com/2011/02/10/camelcase-vs-underscores-scientific-showdown/

- **80 characters per line**
- **Javadoc comments for classes, fields, and methods**

```
/** Description of what the program does
 *
 * @author Sara
 */
```

Javadoc example: class Random

https://docs.oracle.com/javase/7/docs/api/java/util/Random.html

JAVA STYLE CONVENTIONS

- **Whitespace**
  - When in doubt, include a blank line!

JAVA STYLE CONVENTIONS

- **Whitespace**
  - When in doubt, include a blank line!

- **Each imported class called separately**

```
import java.util.Scanner;
import java.util.Random;
```

JAVA STYLE CONVENTIONS

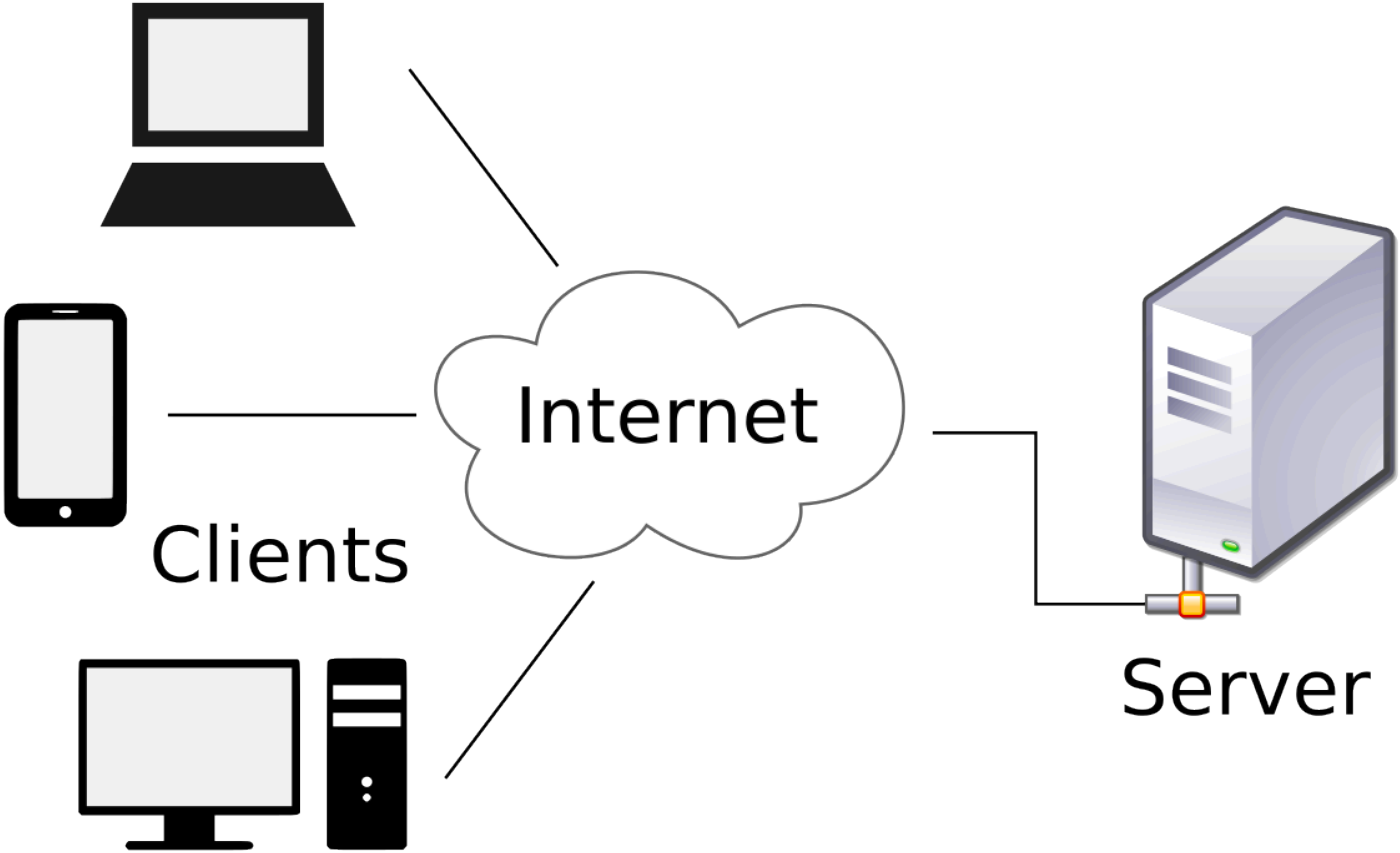
- **Whitespace**
  - When in doubt, include a blank line!

- **Each imported class called separately**

```
import java.util.Scanner;
import java.util.Random;
```

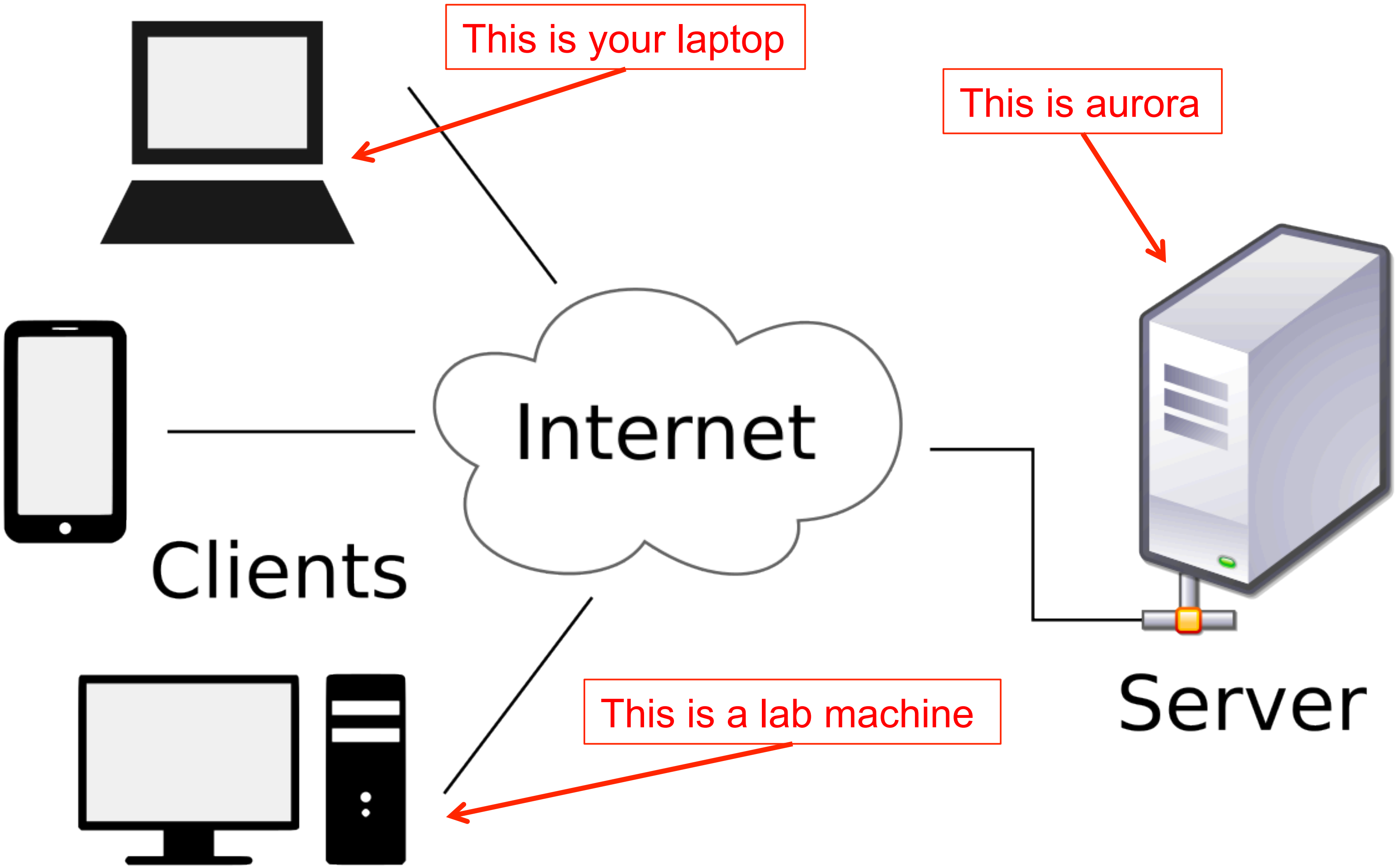
- **Full English names for variables, methods, and classes**
  - Exception: some numbers (i.e. integer *n*, iterator *i*)
  - Example: `userWords` vs. `w`

CLIENT-SERVER MODEL



https://en.wikipedia.org/wiki/Client-server_model

CLIENT-SERVER MODEL



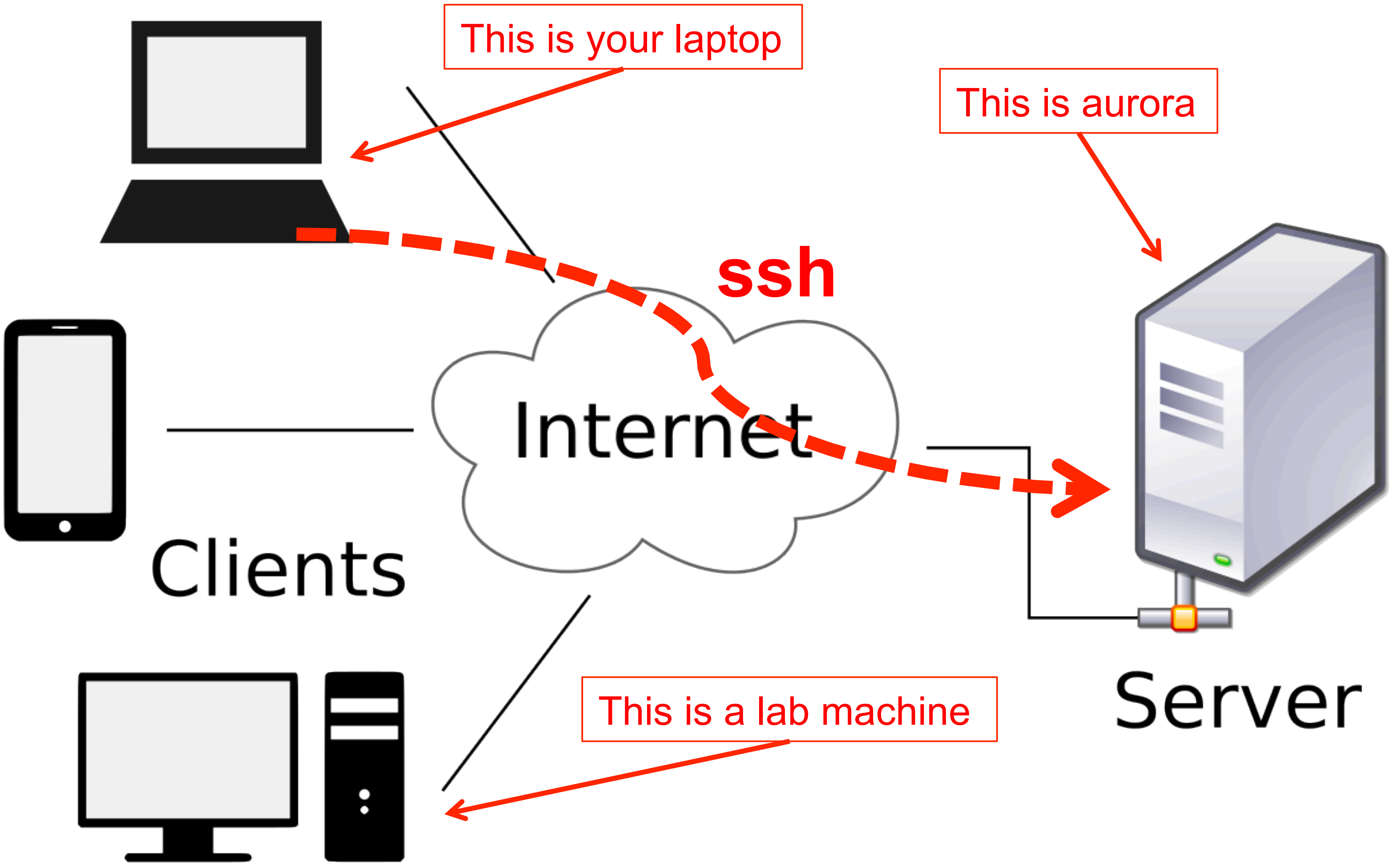
This is your laptop

This is aurora

This is a lab machine

https://en.wikipedia.org/wiki/Client-server_model

CLIENT-SERVER MODEL



This is your laptop

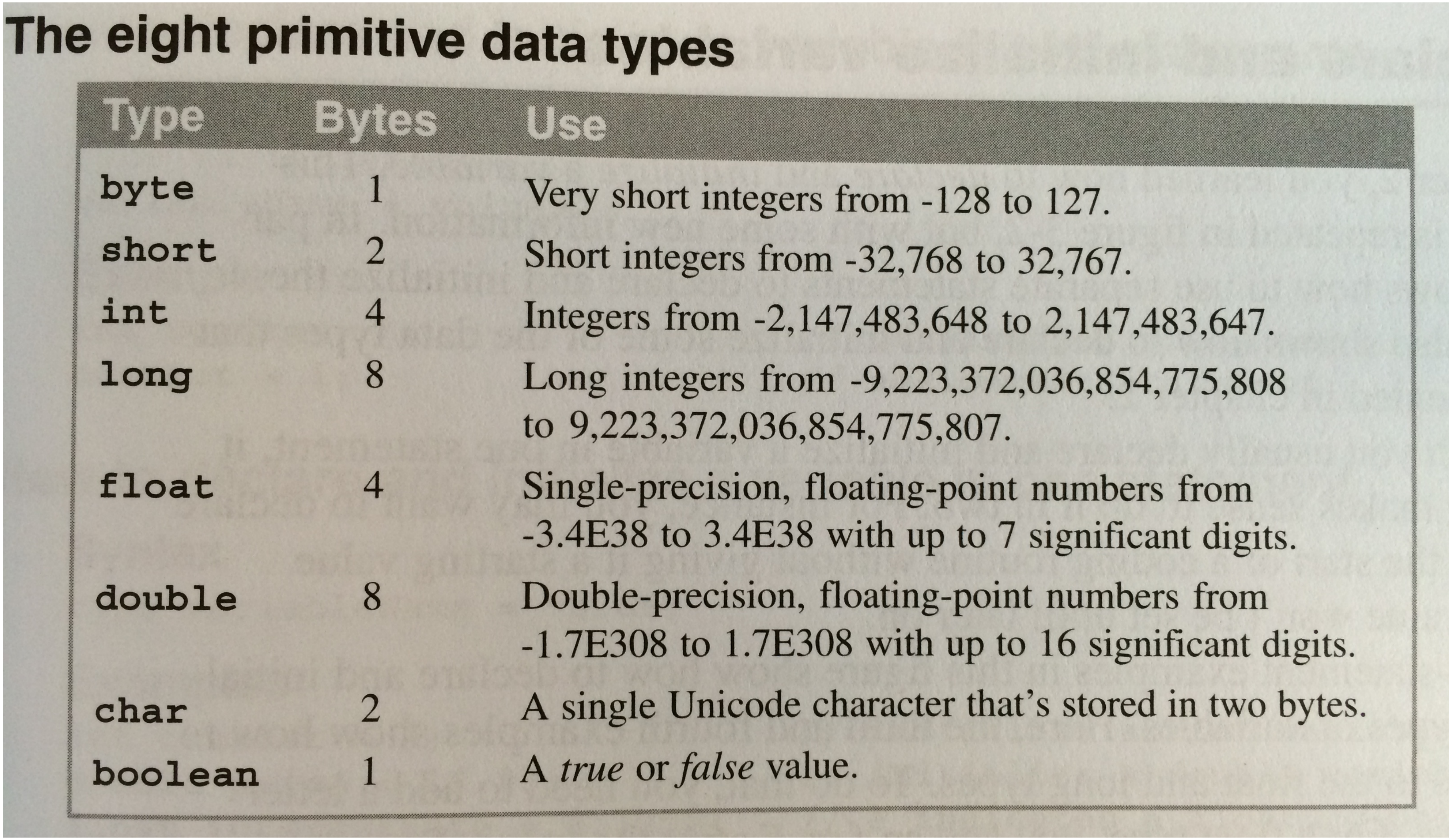
This is aurora

ssh

This is a lab machine

https://en.wikipedia.org/wiki/Client-server_model

JAVA PRIMITIVE TYPES



The eight primitive data types

Type	Bytes	Use
`byte`	1	Very short integers from -128 to 127.
`short`	2	Short integers from -32,768 to 32,767.
`int`	4	Integers from -2,147,483,648 to 2,147,483,647.
`long`	8	Long integers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.
`float`	4	Single-precision, floating-point numbers from -3.4E38 to 3.4E38 with up to 7 significant digits.
`double`	8	Double-precision, floating-point numbers from -1.7E308 to 1.7E308 with up to 16 significant digits.
`char`	2	A single Unicode character that's stored in two bytes.
`boolean`	1	A *true* or *false* value.

(Murach's Java, Chapter 3)

FIRST DATA STRUCTURE

Arrays!