Object-Oriented Programming unit:

- Review of object-oriented concepts (week-before Friday and last week Monday)
- Review of linked lists (last week Wednesday)
- More on linked structures (last week Friday)
- Documentation; Java GUI components (Monday)
- Git; abstract data types (**Today**)
- More abstract data types; Java Collections (Friday)
- Next week: Review (Wednesday) and Test 1 (Friday)

Today:

- Finishing quick sort on lists
- Introduction to git
- Abstract data types

Coming up:

- **Due Thurs, Feb 12, at 10:30am**
  - *Read prelab reading*
  - *Make a github account*
  - *Generate ssh keys*
  - *Take quiz*
- **Due Wed, Feb 18.** *Do Project 2, "First Calculator."*

Summary of git commands:

	Description	Example
clone	Grab a copy of a repository	`git clone gitgithub.com:TVD-WC-CSCI/CSCI245S26_T0.git`
add	Indicate a new or changed file	`get add SomeFile.java`
commit	Commit changes to local repository	`get commit -m "Added method m()"`
push	Send changes to shared repository	`git push`
pull	Grab changes from shared repository	`git pull`
log	Display a record of commits	`git log`
status	Compare working copy with repository	`git status`

- A *type* is a set of value associated by how they are stored in memory and what operations are defined for them.
- An *abstract data type (ADT)* is a type whose values are specified by their logical properties and their operations, not by how they are stored in memory or how the operations are implemented.
- A *list*, as an abstract data type is an ordered collection of items, such that items can be added, retrieved, removed, and processed in order.

Other examples of ADTs:

- A *set* is an unordered collection of items in which no item is in the collection more than once, with the following operations:
  - Test whether the set contains a given item.
  - Test whether the set is empty.
  - Add an item to the set.
  - Remove an item from the set.
  - Iterate through all the items in an arbitrary order.
- A *map* is an association (or mapping) between elements of two types, called the *keys* and *values*, with the following operations:
  - Put a new association between a given key and value.
  - Get the value associated with a given key.
  - Remove a given key and its associated value.
  - Test whether the map has an association for a given key.

Coming up:

- **Due Thurs, Feb 12, at 10:30am** *Read prelab reading, make a github account and ssh keys, and take quiz*
- **Due Wed, Feb 18.** *Do Project 2, "First Calculator."*