One-third-way point of the semester:

- More on linked structures (last week Friday)
- Documentation; Java GUI components (Monday)
- Git; abstract data types (Wednesday)
- More abstract data types; Java Collections (**Today**)
- Review (next week Wednesday)
- Test 1 (next week Friday)
- Object-oriented design and class extension (week after test)

Today:

- ADT definition (review)
- Common Java Collections interfaces and classes
- Differences between Linked-based and array-based implementations
- Revisiting "homemade" versions of ADT implementations

Coming up:

- **Due Wed, Feb 18.** *Do Project 2, "First Calculator."*
- *(There will be a quiz before lab on Thurs, Feb 19, but no prelab reading)*
- **Due Fri, Feb 27.** *Do Project 3, "Homemade Linked-list Map." (Recommendation: Do this by Fri, Feb 20, to help prepare for the test.)*
- **Due Fri, Mar 6.** *Do Project 4, "Text-based adventure game." (You'll have the next three lab periods to work on this. . . don't work on it outside of class yet.)*

- 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.