In-class exercises (lecture 8)

Exercise 1: Practice with GitHub branches

In your own repo, using the command line

```
cd path/to/your/repo
git status
```

If you see a message telling your repo is ahead of the remote repo, commit and push before continuing.

```
git pull
git checkout -b your_branch_name
```

Write some code, commit and push. You will need to set the remote branch on the first push:

```
git push --set-upstream origin your_branch_name
```

Exercise 2: Generic stack and queue

Refactor the stack and queue implementations using generics so that these data structures can store any object. Implementations of both data structures are in the lecture-code repo, starter code for this week. You will need to modify:

- Interfaces
- Implementing classes
- Tests

When you get started, you will probably see compile errors in the existing Stack and Queue implementations. These errors occur because the underling generic linked list is returning a generic type for methods like getItem. These should go away once your Stack and Queue are also generic.

When you run tests, you may also see “unchecked” Xlint warnings. See if you can figure out why these warnings are happening!

Exercise 3: Extend a generic class

Write a HighScore class that extends the Pair class (in the pairs package)

HighScore contains:

- username: a String

- score: an Integer
- Appropriately named getters for each of the above

Exercise 4: Doctors and vets

In genericpatientslist:

- Create a generic abstract class called AbstractProvider to represent a health care provider. The generic type will be used to indicate the patient type of concrete implementations of the abstract class.
- Refactor the Vet class so that it extends AbstractProvider.
- Implement AbstractProvider in a concrete Doctor class.
- Vets and doctors both have a maximum number of patients that they can accept, and a PatientList.
- Use generics appropriately to ensure that instances of the Vet class can only have non-humans on their PatientList and that instances of the Doctor class can only have people on their PatientList