TA: Sherdil Niyaz

# CS61B Extra Problems 2

## Git

Sometimes it can be unclear what state a file is in with Git.

- If you create a brand new file called hug.java in a repo (repository), what state is it in?
- If you then `add` that file, what state is it in?
- Great! You now edit hug.java and decide to `commit` the changes you made to that file. Now what state is hug.java in?
- Finally, you take hug.java and edit it. If you `commit` now, will the changes be included? Why or why not?

## Git: Detached Head

Oh no! You got a `detached head error`! What will you do?

## Box and Pointer

Note: Adapted from Jonathan Shewchuck

(CS61B Fall 2006, Midterm 1)

The following code creates a linked list representing the Fibonacci series in reverse order. (Don't worry if you don't know what that is.) Suppose we execute main. Draw the stack and heap at the moment when the topmost fibonacci call on the stack is about to "return n". Specifically, draw a box-and-pointer diagram with a box for every stack frame, local variable, object, and field in memory at that moment. Include the stack frames for all methods in progress, and illustrate which entities are inside those stack frames. Don't forget "this". Entities on the stack should be on the left-hand side of the page, and entities on the heap (aka objects) should be on the right-hand side.

*See next page

TA: Sherdil Niyaz

```
public class ListNode {
  public int item;
  public ListNode next;

  public ListNode(int i, ListNode n) {
    item = i;
    next = n;
  }

  public ListNode fibonacci(int i) {    /* Appends i more terms to series. */
    /* Next number is sum of previous two numbers in series. */
    ListNode n = new ListNode(item + next.item, this);
    if (i <= 1) {
      // Draw the stack and the heap when execution reaches this point.
      return n;
    } else {
      return n.fibonacci(i - 1);
    }
  }

  public static void main (String[] args) {
    /* First 2 numbers in series are 1. */
    ListNode n = new ListNode(1, new ListNode(1, null));
    n = n.fibonacci(3);                          /* Append 3 more terms. */
  }
}
```