1. (4 points) What is the advantage of getting in the habit of writing simple code?

2. (5 points) Write a fragment of code that would compile but cause a NullPointerException if executed.

3. (5 points) Below is a broken update method for a Conway's Life program. What's wrong?

   How would you fix it?

```
void update() {
     history();
     for (int row = 0; row < size; row++) {
         for (int col = 0; col < size; col++) {
             cells[row][col].setLiveNbrs(countEm(row, col));
             cells[row][col].update();
         }
     }
 }
```

4. (5 points) Write an equals method which is passed the current and previous state of a Conway's Life simulation. It should return true just if they are identical. Assume Cell has a getAlive() method, and that the arrays are NxN. The heading has been provided for your convenience.

```
boolean equals(Cell[][] one, Cell[][] two) {
```

5. (5 points) When is an algorithm, A, with running time on input of length n, $T_A(n)$, said to be O(f(n))? I.e. what is the definition of O(f(n))?

6. (4 points) If an algorithm is O(n), does that mean it is also O(n!)? Explain why or why not.

7. (4 points) What would happen if this code were executed?

```
Foo[] list = new Foo[10];

for (int i=0; i<10; i++)
    System.out.println(list[i]);
```

8. (5 points) Write pseudocode to evaluate an arbitrary postfix expression. Assume you are provided with a working Stack class, an Emitter (that emits whatever you pass it) and a Tokenizer (with `String nextToken()`, and `boolean hasMoreTokens()`).

9. (4 points) Draw the tree representing this expression: 1 + 2 * 3 - (4*5)

10. (5 points) What are the two steps in a proof by induction?

    i

    ii

11. (4 points) When do those two steps together create a proof for all natural numbers? Why?

12. (5 points) You have two sorts which you analyze to be $\theta(n^2)$; you time them and discover that one runs a million times faster than the other. Could they both be $\theta(n^2)$? Explain.

13. (10 points) Write a method, digitAt, that is passed two ints and returns the value of the digit in the first at the index of the second (where the least significant digit is index 0). I.e. assuming the first parameter is 1234; if the second is 0, it should return 4; if the second is 3, it should return 1, and if the second is 4 or more it should return 0.

14. (5 points) Write pseudo-code for selection sort. Indicate with a circle and an arrow, <-----, which portion will execute n^2 times.

15. (5 points) What is the recursive definition of a binary tree (as presented in class)?

16. (5 points) If the input to quick sort or tree sort (using a binary search tree) is in order (either ascending or descending), their running times become n^2; why?

Quicksort:

Treesort:

17. (10 points) Use an `ArrayList<Bar>` to write a `Queue` class with `void enqueue(Bar)`, `Bar dequeue()` and `boolean isEmpty()`.

18. (10 points) Given a `Queue` class, implement `Stack` (with push, pop, and isEmpty()) utilizing 2 `Queues` (no other data structures allowed!). Read 19 before proceeding!

19. (10 xtra credit) Do the previous problem using only one `Queue` instead of 2. In the space above!