1 Boxes and Pointers II

Draw a box and pointer diagram for each code block.

(a) \[
\begin{align*}
&\text{int[]} \ x = (1, 2, 3); \\
&\text{int[]} \ y = x; \\
&y[2] = 7;
\end{align*}
\]

\(x\) and \(y\) should both point to an array with values \([1, 2, 7]\).

(b) \[
\begin{align*}
&\text{IntList} \ l = \text{IntList.list}(1, 2, 3); \\
&\text{IntList} \ l2 = l; \\
&l.\text{tail}.\text{tail}.\text{head} = 7;
\end{align*}
\]

\(l\) and \(l2\) should both point to an \text{IntList} with values \(1, 2, 7\).

(c) \[
\begin{align*}
&\text{IntList[]} \ ll = \text{new IntList}[3]; \\
&ll[0] = \text{IntList.list}(1, 2); \\
&ll[1] = \text{IntList.list}(2);
\end{align*}
\]

\(ll\) should point to an array, where the first two elements point to \text{IntLists} and the third is null.

2 Objects Refresher: Does this make sense?

(a) Determine what would be printed after executing the main method of class \textit{Avatar}.

\begin{verbatim}
public class Avatar {
    public static String electricity;
    public String fluid;

    public Avatar(String str1, String str2) {
        Avatar.electricity = str1;
        this.fluid = str2;
    }

    public static void main(String[] args) {
        Avatar foo1 = new Avatar("one ", "two");
        Avatar foo2 = new Avatar("three ", "four");
        System.out.println(foo1.electricity + foo1.fluid);
        foo1.electricity = "I declare ";
        foo1.fluid = "a thumb war";
        System.out.println(foo2.electricity + foo2.fluid);
    }
}
\end{verbatim}

The main method will print

\begin{verbatim}
three two
I declare four
\end{verbatim}
(b) Consider swapping Avatar and this in lines 6 and 7. Which swaps, if any would cause errors if we tried to compile and run the code?

_Both Avatar and this would work on line 6, but only this will work for line 7. Changing this to Avatar on line 7 will cause a compile-time error because we cannot reference instance variables using a static class reference._

(c) Will adding the following method to class Avatar cause any errors during compilation or execution?

```java
public static String getFluid() {
    return fluid;
}
```

The method will cause a compile-time error because we can not reference an instance variable (in this case, fluid) from inside a static context.

When the object is not specified (the thing before the period) in a field access or method call, Java will use this by default. However, since the new method is static, this does not exist and therefore an error is thrown.

3 Min/Max

Given an array A, return a 2 element array B where B[0] is the minimum element of A and B[1] is the maximum element of A.

```java
public static int[] minMax(int[] A) {
    int maxVal = Integer.MIN_VALUE; // smallest int in Java
    int minVal = Integer.MAX_VALUE; // largest int in Java

    int[] B = new int[2];

    for (int i = 0; i < A.length; i++) {
        maxVal = max(maxVal, A[i]);
        minVal = min(minVal, A[i]);
    }
    B[0] = minVal;
    B[1] = maxVal;
    return B;
}
```
4 Reverse

Given an array A, reverse its elements in place (i.e. do not create any new arrays; this should be a destructive method).

```java
public static void reverse(int[] A) {
    for (int i = 0; i < A.length / 2; i++) {
        int temp = A[A.length - i - 1];
        A[A.length - i - 1] = A[i];
        A[i] = temp;
    }
}
```