Midterm 1 additional problems (do after class)

Here are some additional problems that involve reading code rather than writing it.

1. Imagine these two functions below live inside some Java class, and we run the class (so main() is called). What is the output?

```
public static int func(int[] array, int p) {
    for (int i = 0; i < array.length; i++) {
        array[i] *= p;
    }
    p = array[2];
    return array[1];
}

public static void main(String[] args) {
    int[] nums = new int[] {1, 2, 3};
    int a = 4;
    int y = func(nums, a);
    System.out.println(nums[0] + " " + nums[1] + " " + nums[2]);
    System.out.println(a);
    System.out.println(y);
}
```

   What is the output when main() is called? Hint: this is not supposed to be a trick question; the code doesn't crash.

2. A student has written the code snippet below, which is intended to examine an array of integers to find the smallest number within the array and its corresponding index. The line numbers at the left are for convenience; they are not part of the code itself. You may assume the code below lives inside some main() function in some class, and the array named `array[]` has been properly declared and initialized earlier in the code.

```
1     int min = 0;
2     int minIndex = 0;
3     for (int i = 0; i <= array.length; i++) {
4         if (array[i] < min)
5             min = array[i];
6             minLocation = i;
7     }
8     System.out.println("min=" + min + " and index=" + minIndex);
```

   There are 3 bugs in the program above. Each bug causes either an incorrect answer or crashes the program. Identify each bug and state how to fix them. You can use the line numbers to talk about where the bugs are.

3. Here is sample code for a Cat class:

```
public class Cat {
   private String name;
   private int age;

   public Cat(String newName, int newAge) {
      name = newName;
      age = newAge;
   }

   public int getAge() {
      return age;
   }

   public String getName() {
      return name;
   }

   public void setAge(int newAge) {
      age = newAge;
   }

   public void setName(String newName) {
      name = newName;
   }
```

Now imagine you are given the following code snippet (that lives inside a main() function somewhere). Draw four memory diagrams (the state of the computer's memory, showing each object as a box and all references to objects with arrows) at each point in the program labeled "draw here".

```
Cat cat1 = new Cat("Oliver", 4);
Cat cat2 = new Cat("Tigger", 5);
Cat cat3 = cat1;

// draw here

cat1.setAge(6);
cat3.setName("Simba");
cat3 = cat2;
cat3.setAge(7);

// draw here

Cat[] array = new Cat[4];
array[0] = new Cat("Nala", 3);
array[1] = new Cat("Mufasa", 9);
array[2] = new Cat("Scar", 8);
array[3] = array[0];

// draw here (only the cat array is needed in the diagram)

array[2].setAge(7);
array[3].setAge(4);
array[0].setAge(array[0].getAge() + 1);
array[0] = array[1];
array[1].setAge(array[1].getAge() + 1);

// draw here (only the cat array is needed in the diagram)
```