Homework 3 (10 points)

Apply what you know about class design (what to do, what not to do, and why) to a critique of the Java program included with this assignment, BadClass.java. There are almost as many design don’ts in this file as there are lines of code, so don’t hold back.

What to turn in:

- A few paragraphs describing what, from a design standpoint, is wrong with this class. Some specific questions you might ask (and answer) include:
  - Does this class use encapsulation/information hiding?
  - Does this class follow the given guidelines for accessor and mutator methods?
  - Does this class avoid side effects?
  - Does the class describe a cohesive abstraction? If so, what? If not, why not?
  - Is the class complete in its support of its abstraction (if it supports one)?
  - Is the interface convenient? Is it clear?
  - Does it use a consistent naming scheme?
  - Are pre and postconditions clearly stated?
  - Are there any class design guidelines this class doesn’t violate? If so, which one(s)? If not, you should give examples of each violation (if you haven’t already done so).

- Do not just give yes/no answers to the questions above – give specific examples in the code to support your answers.

- If you wish, you may turn in your assignment as an annotated version of the original source code. A possible format is given below:

```java
/**
 * This class compiles! The code runs! So why does it suck?

 *** Student critique here: this class sucks because the comments ***
 *** includes the word “suck,” which is offensive, and sucks. ***

 *
 * @author Cate Sheller (but she will deny it profusely)
 * @version -0
 */
```

Your own annotations would, of course, be far more professional (not to mention to the point, which the above is not).