Solution to Practice Problem: 
Class Debugging

**Debugging programs:** Identify as many errors as possible in the following code. Circle the part of the code that causes the error, draw an arrow to it, and then describe as precisely as possible what the error is.

```java
public class Mystery1
{
    private int x = 0;
    private String s;
    private static int y = 3;

    public Mystery1(int x, String s)
    {
        this.x = x;
        s = this.s;
    }

    public static void update(int a, String b)
    {
        x += a;
        s += b;
    }

    public void toString()
    {
        return s + y;
    }
}

public class Mystery2
{
    public static void main(String [] args)
    {
        Mystery1 m = new Mystery1();
        m.x = 3;
        m.update(4, "hello");
        System.out.println(Mystery1.toString());
    }
}
```

Comment [APY1]: This is not a compile-time error, but a logic error. Most likely, the programmer intended to say `this.s = s;` which is a way of initializing the field `this.s` using the constructor parameter `s`. The way it is now, it sets the parameter to be the initial value of `this.s` (which is auto-initialized by Java to be null). That doesn't really accomplish anything.

Comment [APY2]: Cannot refer to instance fields `x` and `s` from a static context. To fix, make this an instance method (by removing the keyword "static").

Comment [APY3]: Overrides the `toString` method from the Object class, but with a different return type (void instead of String). Alternatively, you can say that the error is that the method is declared to be void, but the body of the method returns a String.

Comment [APY4]: This is NOT an error. It is totally fine to refer to a static field from an instance method/context. It's the reverse situation that's a problem.

Comment [APY5]: Wrong number of arguments for the `Mystery1` constructor

Comment [APY6]: `x` is private in class `Mystery1`

Comment [APY7]: This one looks like an error, but it is NOT. It is fine to call a static method using a reference variable. However, it is recommended to call a static method using the more normal way, which is to use the name of the class instead of the instance: `Mystery1.update(4, "hello").` As above, however, the correct fix here is to leave the method call the way it is, and instead to change the definition of the update method to be an instance method, since the body of the method refers to instance fields.

Comment [APY8]: Cannot call an instance method from a static context. To fix, you would need to replace "Mystery1." with a reference variable of type `Mystery1`. For instance, "m.toString()" would be perfectly fine.

Comment [APY9]: A second error here: `toString()` is currently declared to be void. You cannot use a void method as part of a larger expression (in this case, as the argument of `println`). As with the comment above, the way to fix this is to change the return type of `toString` to be "String" instead of "void".