4.Look at the following class:

```
1   public class Test {
2       private String testName;
3
4       public Test( String name ) {
5           this.testName = name;
6       }
7
8       public setTestName( String name ) {
9           this.testName = name;
10      }
11  }
```

What would be the proper way to construct a Test object with member variable testName initially being "old", then later changed to "new"

- ○
```
1   Test testName = "old";
2   testName = "new";
```

- ○
```
1   Test testObj = new Test( "old" );
2   testObj.testName = "new";
```

- ○
```
1   Test testObj = new Test( "old" );
2   testObj[testName] = "new";
```

- ✓
```
1   Test testObj = new Test( "old" );
2   testObj.setTestName( "new" );
```

3. Look at the following code:

```
int errorInteger = 200;
String comment;

switch (errorInteger) {
    case 150:
        comment = "Javascript error.";
    break;
    case 240:
        comment = "Comment error.";
    break;
    case 300:
        comment = "Function error.";
    break;
    case 200:
        comment = "New error.";
    break;
    default:
        comment = "No error.";
    break;
}
System.out.println( comment );

```

What would be the resulting output from this code?

- ○ Comment error.
- ○ New error.
- ○ Javascript error.
- ○ Function error.

What is defined in the denoted sections of this class?

- ○ **section 1:** member variable
  section 2: constructor
  section 3: class method
- ○ **section 1:** member variable
  section 2: class method
  section 3: method
- ○ **section 1:** method
  section 2: constructor
  section 3: member variable
- ✓ **section 1:** member variable
  section 2: constructor
  section 3: method

---

2. As an established Java convention, what would it mean if the name of a variable was spelled in all uppercase?

- ○ Nothing. There is no such convention, and such a variable is like any other.
- ✓ e variable is a constant, whose value should not change.
- ○ The variable is contains a string that has all capital letters.
- ○ The variable is reserved for use by the Java environment, and you should not refer to it.