

CS139 – Coding Style



Programming Poorly

```
1 import java.util.Scanner;
2     public class Sec {
3     public static void main(String[] args) {
4         int s;int h6;
5         int M;
6         int s2;
7         Scanner k; k = new Scanner(System.in);
8         System.out.print("Enter input: ");
9         s = k.nextInt();
10            s=s +0+ 0 + 0;
11 System.out.println();
12 h6=s /3600; M=s %3600/60;s =s+0; s2=s %3600%60;
13     System.out.println(
14
15 s+ " s = "+ h6 + " h, "
16 + M +" m, and "+ s2 + " s\n"); s = s2;
17     }}
18
```

Elements of Style

- Consistent formatting
 - Reading code is hard enough *without* unfamiliar formatting
 - Maintaining consistency inside an existing code base is more important than enforcing your own sense of style
 - Formatting can be checked with automated tools

This Code Passes Automated Formatting tests...

```
1 import java.util.Scanner;
2
3 /**
4  * My AWESOME PROGRAM.
5  * @author Nathan Sprague
6  * @version 2/1/2016
7  */
8 public class Walterbot {
9
10     /**
11     * Fill this in later.
12     * @param args Not used
13     */
14     public static void main(String[] args) {
15         Scanner readerator;
16         readerator = new Scanner(System.in);
17         System.out.print("Enter the number of seconds: ");
18         int larry; // integer
19         larry = readerator.nextInt();
20         int curly; // integer
21         int beep; // Another integer!
22         beep = larry % 3600 % 60; // MOD MOD MOD
23         System.out.println(); // Print a newline character
24         int moe; // integer
25         moe = larry / 3600; // DIVISION
26         curly = larry % 3600 / 60; // DIVISION AND MOD
27         System.out.println(larry + " seconds = " + moe + " hours, "
28                             + curly + " minutes, and " + beep
29                             + " seconds\n");
30     }
31 }
32
```

Is this
good
style?

Programming Well...

Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

The practitioner of literate programming can be regarded as an essayist, whose main concern is with exposition and excellence of style. Such an author, with thesaurus in hand, chooses the names of variables carefully and explains what each variable means. He or she strives for a program that is comprehensible because its concepts have been introduced in an order that is best for human understanding, using a mixture of formal and informal methods that reinforce each other.

Donald Knuth. "Literate Programming (1984)"

Automated tools cannot check “excellence of style”. It takes practice.

