

Evaluating the Style of your programs

Objectives

At the end of this exercise, students will:

- Be able to evaluate a program for conformance to style guide for the class.
- Be able to evaluate arithmetic expressions in Java

Roles

- Coordinator – will make sure the team stays on task
- Presenter – will only be used if there is a question about your group's responses
- Recorder – will write on the board the group answers to BOARD questions
- Reflector – if present will fill out the exit pass, otherwise, team members will do so

Getting ready

1. All team members - record your notes on your own response sheet.

Part 1 – Why a Style Guide

1. Review the three Payroll.java solutions. Each was written by a different person and have different levels of quality. DO NOT WORRY about the Style Guide yet.
2. Which of the three do you consider to be the “best” solution? Why? Note that all passed submit tests, so are all considered “correct” solutions.
3. Which of the three do you consider to be the “worst” solution? Why?
4. In your own words, what are the qualities of a good program from a stylistic point of view?

Part 2 – Evaluating a program

5. Using **PayrollV3**, apply the evaluation rubric to the program. Specifically,
 - a. Items to look for are grouped into general areas, such as A. Identifier naming. Subitems indicate a particular feature that we require. So for example, if you see a name like Q for a variable name, it violates both A.1 and A.2 since variables must begin with a small letter and be meaningful.
 - b. For each item in the list, look to see that the program conforms to that item. For each violation, write the code corresponding to the problem beside the issue in the program listing (like A.1).
 - c. Proceed through the program indicating all such violations.
 - d. On the evaluation sheet, count the number of violations and indicate that number in the blank box to the right of the described issue. So for example, if there were two identifiers that did not have meaningful names, the space beside A1 would be a 2. If you find more than 2 violations, the maximum deduction is 2. For any major category, the maximum violations are 5.
 - e. Compare your answers within the team and have the reporter ready to respond when / if your team is called.
 - f. **BOARD** – Given that 20 points will be devoted to Style Guide issues, how many of the 20 points would this program receive? Identify the first 5 errors you found by code number.



STOP HERE

6. Give (or show) your SecondsToHours program listing to one other person on the team (or to the other pair).
7. Using another Program Evaluation sheet, look for each violation in your teammate's work and indicate the score for each line. Do not write on the listing as you did for question 8. Write the final score (out of 20) for that program.
8. Switch back so that each person has his/her program back, but not the evaluation sheet. Score your own program, this time indicating violations on your own program listing (if hardcopy) and the score on the evaluation sheet.
9. Compare the paired evaluation sheets. Did you find anything in your own code that your teammate did not? Did you find something in your teammate's code that was a problem that they did not spot? (Don't answer on paper, discuss).

Part 3 – Homework

10. Now that you are more familiar with the conventions, rework your SecondsToHours program and resubmit it. Upload your pdf report to the Assignment that you will find in BB. This needs to be completed before class time on Wednesday.

Part 4 – Operations practice First individually fill in the chart below with the result of each of the Java expressions. Write INVALID in both boxes if there is an invalid expression (one that will not compile). All decimal calculations should be carried out to 2 places only. Check your work with your colleagues.

Expression	Data Type of Result	Result
3 % 13		
13 / 3		
(int) (6.5 / 5)		
(5 % 2) * 7.1		
"abc" + 3 + 17		