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 – Oh goody, another quiz

1. Each person should take the quiz individually, making sure that your name is on top of the paper and that you record your answers on the question sheet. DO NOT USE A CALCULATOR.
2. When everyone has finished, take the group quiz using the IF-AT form. You may grade your own form. 6 points for one scratch off, 3 points for 2, 1 point for 3 and of course 0 for 4. Sum the score and put it on the top of the form. DO NOT grade individual quizzes.
3. As a class, be prepared to ask for clarification of any of the responses.

Part 2 – Why a Style Guide

4. Review the three StyleGuide.java solutions. Each was written by a different person and have different levels of quality. DO NOT WORRY about the Style Guide yet.
5. 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.

6. Which of the three do you consider to be the "worst" solution? Why?

7. In your own words, what are the qualities of a good program from a stylistic point of view?

Part 3 – Evaluating a program

8. 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, indicate on the the problem in the program listing by writing the code (like A.1) beside the problem.
    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. 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?

STOP HERE

9. Give your SecondsToHours program listing to one other person on the team.
10. 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.
11. 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 and the score on the evaluation sheet.
12. 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 4 – Homework

13. 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.

```
 1  import java.util.Scanner;
 2
 3  /**
 4     This program calculates gross pay.
 5
 6     @author Tony Gaddis
 7     @version V1 - 12/15/2007
 8  */
 9
10  public class Payroll
11  {
12     /**
13        main method - entry point to the application
14
15        @param args command line arguments - unused
16     */
17     public static void main(String[] args)
18     {
19        final double BONUS = 500.00;  // bonus amount
20        String name;                  // To hold a name
21        int hours;                    // Hours worked
22        double payRate;               // Hourly pay rate
23        double grossPay;              // Gross pay
24        Scanner keyboard;             // input
25
26        // Create a Scanner object to read input.
27        keyboard = new Scanner(System.in);
28
29        // Get the user's name.
30        System.out.print("What is your name? ");
31        name = keyboard.nextLine();
32
33        // Get the number of hours worked this week.
34        System.out.print("How many hours did you work this week? ");
35        hours = keyboard.nextInt();
36
37        // Get the user's hourly pay rate.
38        System.out.print("What is your hourly pay rate? ");
39        payRate = keyboard.nextDouble();
40
41        // Calculate the gross pay.
42        grossPay = hours * payRate;
43        grossPay = grossPay + BONUS;
44
45        // Display the resulting information.
46        System.out.println("Hello " + name);
47        System.out.println("Your gross pay is $" + grossPay);
48     }
49  }
```

```
 1  import java.util.Scanner;
 2  /**
 3     @author Duke Dog
 4     @version V2 – 09/19/2011
 5  */
 6  public class PayrollV2
 7  {
 8  public static void main(String[] args)
 9  {
10        final double b = 500;
11           String n;
12        int h;
13           double p;
14       double g;
15     Scanner key;  // input
16  key = new Scanner(System.in);
17  System.out.print("What is your name? ");
18        n = key.nextLine();
19  System.out.
20  print("How many hours did you work this week? ");
21  h = key.nextInt();
22  System.out.print("What is your hourly pay rate? ");
23  p = key.nextDouble();
24  g = h*p+b;
25  System.out.println("Hello "+n);
26  System.out.println("Your gross pay is $"+g);
27  }}
```

```
1   import java.util.Scanner;
2
3   /**
4      This program demonstrates the Scanner class.
5
6      @author Duke Dog
7      @version V1 - 12/15/2007
8   */
9
10  public class PayrollV3
11  {
12  /**
13     main method - entry point to the application
14
15     @param args command line arguments - unused
16  */
17  public static void main(String[] args)
18  {
19     final double bonus = 500.00;  // bonus amount
20     String Name;                  // To hold a name
21     int h;                        // Hours worked
22        double payRate;            // Hourly pay rate
23     double grossPay;              // Gross pay
24     Scanner keyboard = new Scanner(System.in);
25
26     // Get the user's name.
27     System.out.print("What is your name? ");
28     Name = keyboard.nextLine();
29
30     // Get the number of hours worked this week.
31     System.out.print("How many hours did you work this week? ");
32     h = keyboard.nextInt();
33
34     // Get the user's hourly pay rate.
35     System.out.print("What is your hourly pay rate? ");
36     payRate = keyboard.nextDouble();
37
38     // Calculate the gross pay.
39     grossPay=h*payRate;
40     grossPay=grossPay+bonus;
41
42     // Display the resulting information.
43     System.out.println("Hello " + Name);
44     System.out.println("Your gross pay is $" + grossPay);
45     System.out.println("Thank you and have a wonderfully beautiful day, " + Name + ".");
46  }
47  }
```