Grade 11 Assignment #5 Java Script

Sleep Debt Calculator

In this program, you are tasked with the job of Calculating if a person is getting enough sleep each week.

Objective: We are going to create a calculator (write a program that calculates) that calculates sleep debt. The program will get how many hours of sleep is ideal each night, then compare that to the actual hours that a person slept each night of the last week.

Then, it will calculate the amount of hours over or under they are to their sleep goal.

Tasks:

There are 15 Tasks to complete for this assignment:

1. Using NotePad++, create the code needed for the program called Sleep Debt Calculator.

   Like in your last assignment, the main part of the program code will only have 1 line in it; a call to a function to calculate one's sleep debt. Only functions generate the code or instructions necessary for the computer to execute in order to make the necessary calculations. The parameters defined by the functions allow arguments to be passed and results returned.

This assignment is truly functional as there are no global variables used, the scope of a program.

2. Create a function. The first problem to solve is asking the user how many hours of sleep they got each night of the week.

   You can declare a function that returns any given night's number of hours of rest.

   Instead of writing 7 functions (one for each day of the week), let's write one function with a parameter for the day.

   Declare a function named getSleepHours with a parameter named day.

3. Inside the getSleepHours function, use either a switch statement to return the number of actual sleep hours for each night.

   For instance, if you got 8 hours of sleep on Monday night, calling getSleepHours('monday') should return 8.

   The function should be able to match any one of the seven days and return a number for each.

4. Now that you've written a function to get the sleep hours for each night, we need to do three things:
   - Get the total sleep hours that the user actually slept
   - Get the ideal sleep hours that the user prefers
   - Calculate the sleep debt, if any.

   To get the total sleep hours that the user actually slept, declare a new function named getActualSleepHours that takes no parameters.

5. Inside the getActualSleepHours function, call the getSleepHours function for each day of the week. Add them all together and return the result.

6. To get the ideal sleep hours that the user prefers, declare a function named getIdealSleepHours with no parameters.

   Inside the function, declare a variable named idealHours and set its value to your ideal hours per night. Then return the idealHours multiplied by 7.

   You'll want to multiply by 7 to get the total hours you prefer per week.

7. Now that you can get the actual sleep hours and the ideal sleep hours, it's time to calculate sleep debt.

   Declare a function named calculateSleepDebt with no parameters.

   Inside of its block, create a variable named actualSleepHours set equal to the getActualSleepHours function call.

   Then, create another variable named idealSleepHours, set equal to the getIdealSleepHours function call.

8. Now that you have actualSleepHours and idealSleepHours, you can write a few if/else statements to output the result to the user. The function should fulfill this logic:
   - If actual sleep equals ideal sleep, log to the console that the user got the perfect amount of sleep.

- If the actual sleep is greater than the ideal sleep, log to the console that the user got more sleep than needed.
- If the actual sleep is less than the ideal sleep, log to the console that the user should get some rest.

9. To make this calculator more helpful, log the amount of hours the user is over or under their ideal sleep in the calculateSleepDebt function's output.

10. On the last line of the program, start the program by calling the calculateSleepDebt function.

11. Just before you call the calculateSleepDebt function (above), print-out a message to the screen welcoming the user and asking them to try out the program while informing them the point of the program.

12. It would be helpful to describe to other developers what this small Java program does. Write some comments that describes what this program does.

    a. Use multi-line comments to (/* comment in the middle of */):
        i. Write one at the top of the code (before the main part of the code) that gives a quick intro/description the assignment.
        ii. Below that comment, write one that summarizes the program.
    b. Use a single line comment to (//then comment):
        i. Create 3 comments anywhere you deem necessary or important in the code. Remember the comment is to highlight or explain what is going on or what is being done in a particular way or used and why.
        ii. Identify the last line of code in every function by writing “End of BLAH-BLAH function”.
        iii. Identify the end of the program.

13. Once your program is complete, make sure to test it using the console (do not submit a program that does not work).

14. Lastly, upload (drag and drop) your assignment to the portal. Look for your name under the Assignment 5 webpage.