Lab 4 - Free Time Calculator

CSE 174 – Fall 2021

lab4: 20 points – Due Saturday, Sep 18, 2021 by 11:59pm

Calculating Free Time

The major responsibility of students is to attend class and complete coursework in a timely manner. Any time not spent sleeping or doing coursework can be considered ‘free time’. While some classes have more or less credit hours, a majority of computing classes are 3 credit hours.

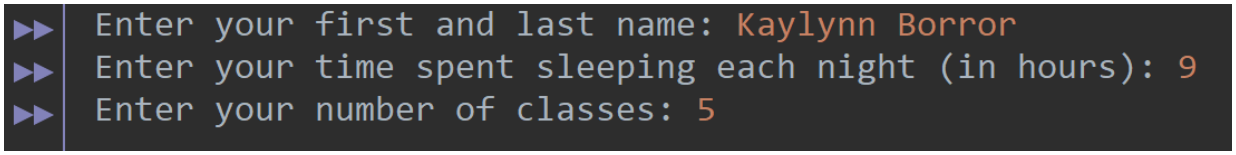
To calculate estimated free time after sleep and class duties (assuming all classes are 3 credit hours), the below equation can be used.

$$freeTime\,(hours) = 120 - (\,hoursOfSleep \times 5 + numClasses \times 8.72\,)$$

In this part of the lab you are to write a program that gets some data from the user and a file, then it calculates the approximate free time based on the given data, and prints the result on the console display and in a file.

Part 1- Reading values from the User

1. Create a class called **Lab4.**
2. Your code needs to get the following values from the user and save them inside proper variables.
    a. First and last name
    b. Hours of sleep (per night)
    c. Number of classes
3. Before you get the values from the user, you need to prompt the user with some messages in each step. **In this assignment you are only allowed to use printf instead of println, or print. If you do use print or println, you will lose points!**



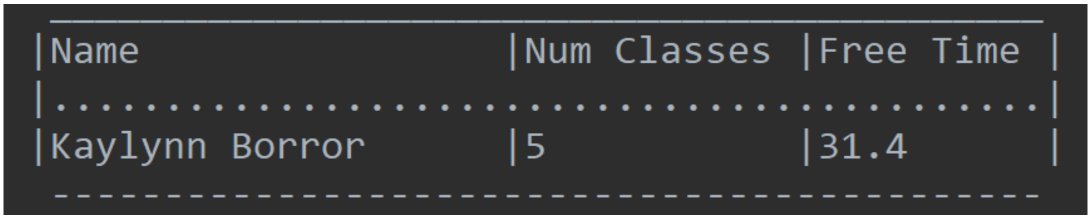
```
Enter your first and last name: Kaylynn Borror
Enter your time spent sleeping each night (in hours): 9
Enter your number of classes: 5
```

4. Now that you have all the values needed, calculate the approximate free time based on the formula shown at the beginning of the instructions, and save the result in a variable.

Instructions Continue on Next Page!

Part 2 - Printing the results on the Console

1. By now you should have all the values and the result inside variables. Now, it's time to print the result on the display, but not as usual. You need to create a table **using printf** and print the result inside the table. So, your result should look like the following:

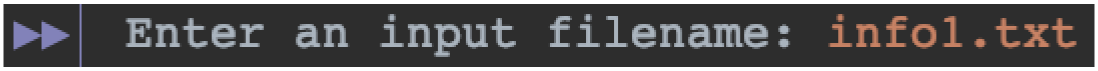


```
 ____________________________________________
|Name                |Num Classes |Free Time |
|............................................|
|Kaylynn Borror      |5           |31.4      |
 --------------------------------------------
```

2. The border at the top is made of underscore characters (_).
3. For printing the first column use **20 characters**, **left aligned**.
4. For printing the second column, use **12 characters**, and the third column uses **10 characters**, both **left aligned** again.
5. The second border between the titles and values is made of period characters (.).
6. The border at the bottom is made of dash characters (-).
7. The free time value is formatted so that there is only **one digit after the decimal point**.
8. **<u>In this assignment you are only allowed to use printf.</u>** So, format your output to make it exactly like the image above.

Part 3 - Reading from the File

1. Download the text file: info1.txt. And put it in the same folder as your java code.
2. Keep your code from part1 and part2 and continue the code by writing a code that gets a filename from the user as follows:

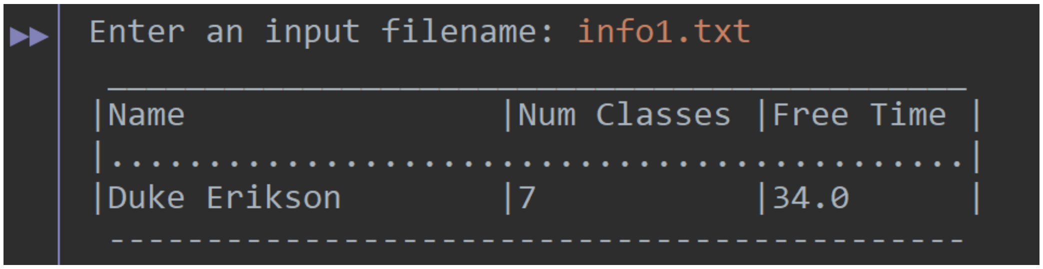


```
Enter an input filename: info1.txt
```

3. Now instead of reading data from the user, create a new scanner variable/object to read the same values from the given filename:
    a. First and last name
    b. Hours of sleep (per night)
    c. Number of classes
4. After reading all the values from the given file and saving them inside variables, now it's time to calculate the free time the same way you did before.

Instructions Continue on Next Page!

5. Print the result on the display using the same table/format that you used before. So, your result should look like the following:

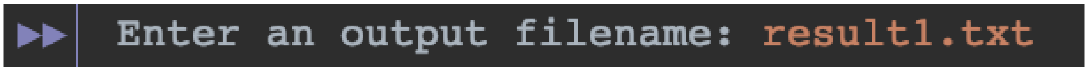


```
Enter an input filename: info1.txt
 ____________________________________________
|Name                |Num Classes |Free Time |
|............................................|
|Duke Erikson        |7           |34.0      |
 --------------------------------------------
```

6. Close the Scanner object that is connected to the file.

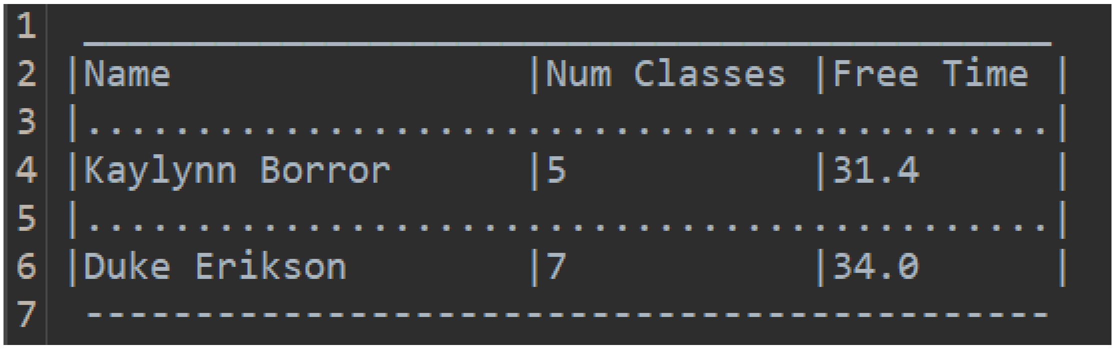
Part 4 - Writing the result inside a File

1. Continue the code and get another filename from the user again:



```
Enter an output filename: result1.txt
```

2. Now, you need to combine the tables from Part2 and Part3 and create the following table as the result. But this table should be written inside the given file, not on the console display. The following is what needs to be written inside the result1.txt file:



```
 ____________________________________________
|Name                |Num Classes |Free Time |
|............................................|
|Kaylynn Borror      |5           |31.4      |
|............................................|
|Duke Erikson        |7           |34.0      |
 --------------------------------------------
```

Tips on Next Page!

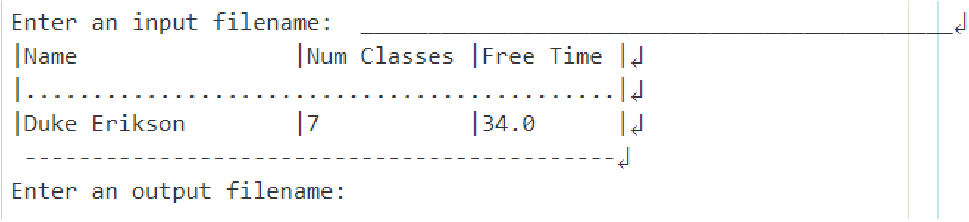
Double check the Following:

- Always use **meaningful** variable names. Don't use names such as "a", "b", "c".
- Always pick the best variable **types** for the variables. If you need the decimal point, use a data type that has one.
- Add comments at the top of your code with at least your name, section, and some descriptions.
- Add comments inside your code to explain what is happening inside the code.
- Remember to follow style guidelines as you write your code.

Submission (You want to actually read this!)

When you believe that you have correctly solved the problem, and that your code follows the style guidelines, submit it on canvas for "Lab 4 - Free Time Calculator". If the autograder rejects your solution, fix it and resubmit it again.

In the **Code plugin you can't see the given inputs along with outputs**, and for that reason the expected output might look a little off compared to your output when you run your code on jGRASP. The following image shows the output in Code plugin:

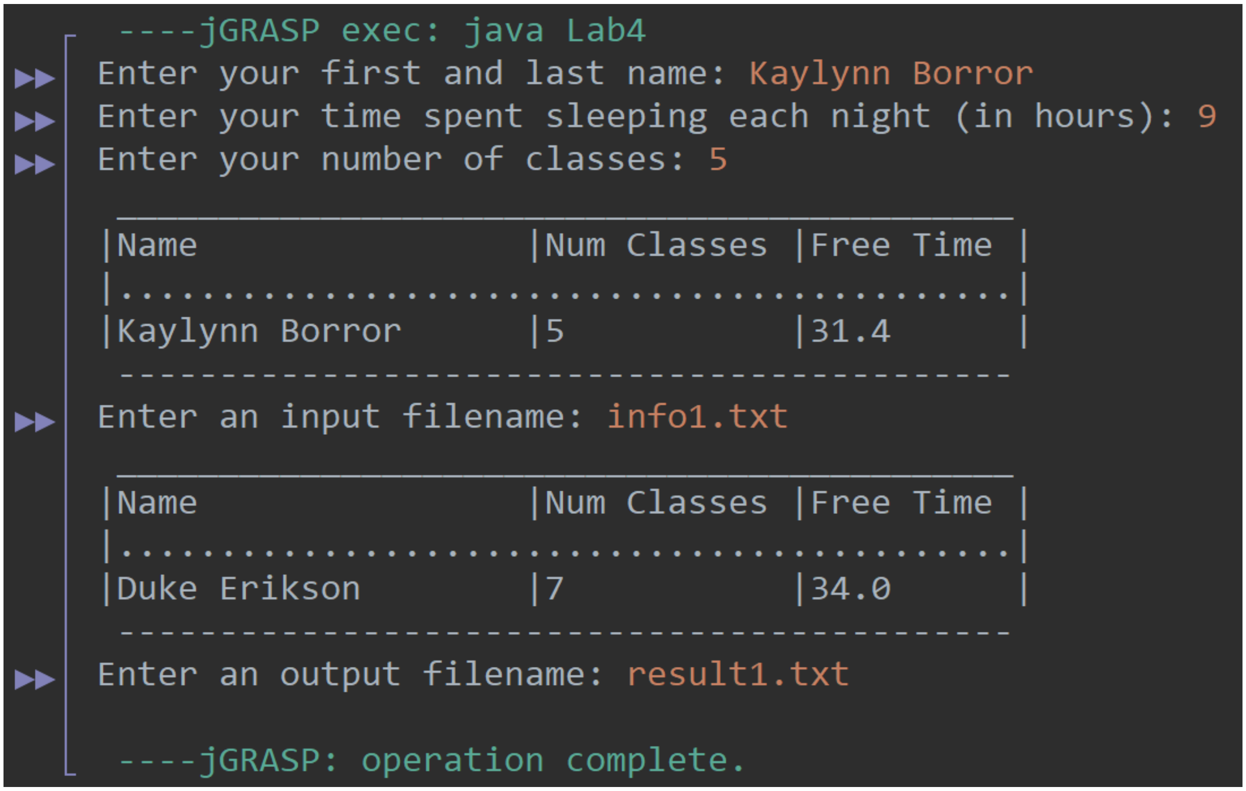


```
Enter an input filename: ____________________________________________↲
|Name                |Num Classes |Free Time |↲
|............................................|↲
|Duke Erikson        |7           |34.0      |↲
 --------------------------------------------↲
Enter an output filename:
```

As you can see after prompting the user to enter a filename, the input does not appear in the expected output and instead, the next line starts right at the same line after that message. This expected output does not mean that your code is wrong or you need to change your code to generate the output like this. Your code still needs to generate the output as shown inside the instructions. It's just the way Code Plugin works, you will get used to it.

Sample Runs on Next Page!

Sample Runs:

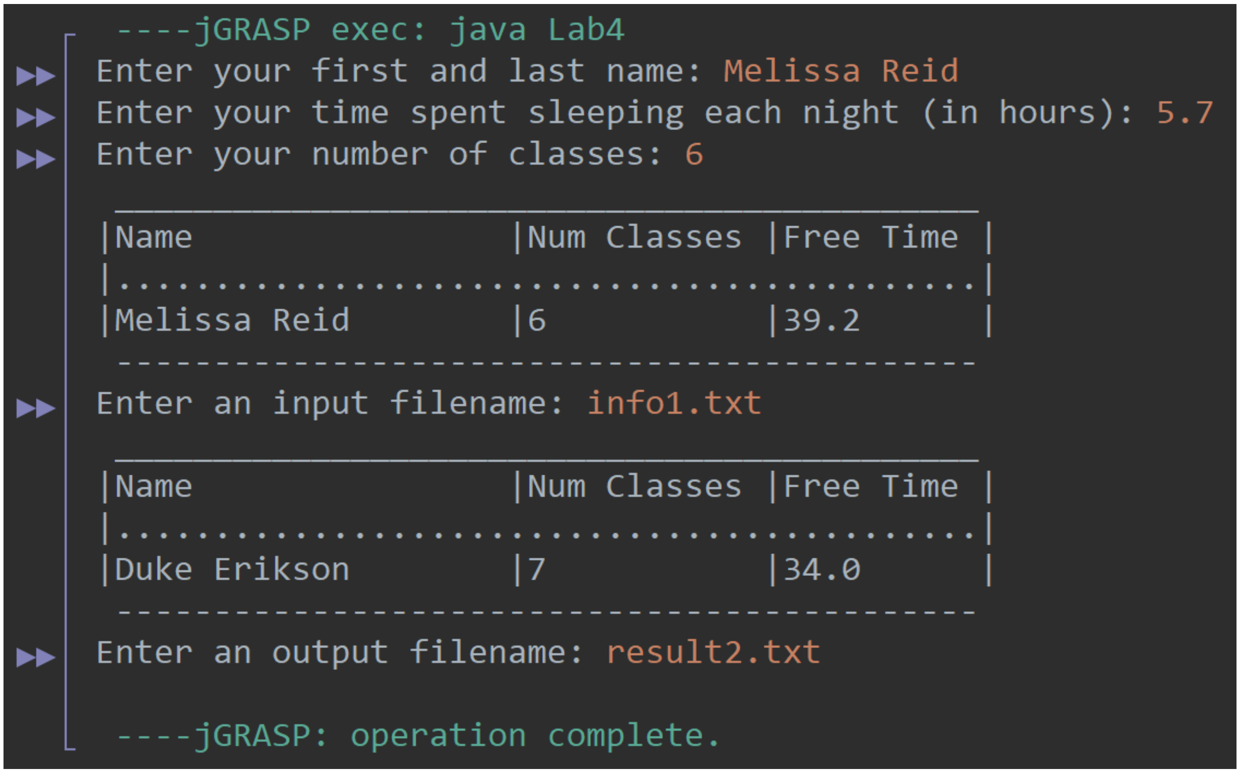


```
 ----jGRASP exec: java Lab4
Enter your first and last name: Kaylynn Borror
Enter your time spent sleeping each night (in hours): 9
Enter your number of classes: 5

 ____________________________________________
|Name                |Num Classes |Free Time |
|............................................|
|Kaylynn Borror      |5           |31.4      |
 --------------------------------------------
Enter an input filename: info1.txt

 ____________________________________________
|Name                |Num Classes |Free Time |
|............................................|
|Duke Erikson        |7           |34.0      |
 --------------------------------------------
Enter an output filename: result1.txt

 ----jGRASP: operation complete.
```



```
 ----jGRASP exec: java Lab4
Enter your first and last name: Melissa Reid
Enter your time spent sleeping each night (in hours): 5.7
Enter your number of classes: 6

 ____________________________________________
|Name                |Num Classes |Free Time |
|............................................|
|Melissa Reid        |6           |39.2      |
 --------------------------------------------
Enter an input filename: info1.txt

 ____________________________________________
|Name                |Num Classes |Free Time |
|............................................|
|Duke Erikson        |7           |34.0      |
 --------------------------------------------
Enter an output filename: result2.txt

 ----jGRASP: operation complete.
```

Rubric on Next Page!

Scoring Rubric:

- **Program produces correct output and follows style guidelines (10 points):** You will receive full credit for this if the autograder accepts your submission. Otherwise your score will be zero.
- **Program only uses printf (2 points):** Your code only uses printf for the whole lab.
- **Program writes inside the given file (4 points):** Your code can write the results inside the given file.
- **Program uses variables wisely (2 points):** Your code uses meaningful variable names and less number of variables and reuses them as much as possible.
- **Program has comments at the top and inside the code (2 points):** Your code has comments at the top with your name and description, and inside the code you have comments explaining the code.