Gym Workout Helper

Documentation

Mateusz Krawczyk

2019

Table of Contents

[1. Requirements – analysis 1](#_Toc26747162)

[1.1. Project brief 2](#_Toc26747163)

[2. Design 2](#_Toc26747164)

[2.1. Flowcharts 2](#_Toc26747165)

[2.1.1. Adding new set 3](#_Toc26747166)

[2.1.2. Adding new exercise 4](#_Toc26747167)

[2.1.3. Calculating recommended training 5](#_Toc26747168)

[2.2. GUI 6](#_Toc26747169)

[2.2.1. GUI layout 6](#_Toc26747170)

[3. Testing 7](#_Toc26747171)

[4. Critique 9](#_Toc26747172)

# 1. Requirements – analysis

* Code must be designed – design will allow reducing numbers of errors in the program and will help with coding the program faster because the program will be well thought out before starting programming.
* The program should show evidence of using loops – loops are useful when you need to run the same code a few times. Loops can run a specified number of times e.g. 10 times, or until a certain condition is met e.g. it will run until the user enters 0.
* The program should show evidence of using branching – branching or if statements mean that the program will execute only some part of code that is chosen based on specified condition e.g. different code will be executed when the user enters 3 and different code will be executed when the user enters 1. This is very useful because you can set conditions in your code that will be executed only if the value as intended.
* The program should show evidence of using file handling – working with files can be very useful for the program. This is because the user can save their progress and the program can read and display the user's progress when they open the program again. Without files, data would be lost as it is stored in the memory and when the device or program is closed, the data is lost.
* The program can make use of existing libraries – using existing libraries can save programmers a lot of time and make their program better, because instead of spending time on something that is already created, the programmer can spend that time on the program and make sure that all functionalities are working correctly.

## 1.1. Project brief

* User can input new exercise or choose one that exists
* User will input how many sets, the weight that they used and how many reps they did in each set
* This data will be saved in a file so the user can access it in the future and see their progress
* With the data, the date of the workout will be saved as well (accessed from the device – datetime library)
* This will be repeated for each exercise that the user selects
* The program will suggest the next workout volume (how many reps and what weight user should use) based on the previous workout
* The program will have a graphic user interface by using the Tkinter module

# 2. Design

Design documents such as flowcharts and GUI designs

## 2.1. Flowcharts

Flowcharts show the logic of the program and the flow of the program as well as any decision making and branching

### 2.1.1. Adding new set



### 2.1.2. Adding new exercise



### 2.1.3. Calculating recommended training



## 2.2. GUI

GUI design shows how the program will look like for the user e.g. layout of buttons and text boxes.

### 2.2.1. GUI layout

USER INPUT

DROPDOWN exercise

BUTTON add exercise

LABEL Add new exercise

LABEL Select your exercise

LABEL Your set info:

LABEL Weight:

USER INPUT

USER INPUT

LABEL Reps:

BUTTON Recommended training

BUTTON Add set

TEXT FILED for displaying workout history or recommended training

# 3. Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No. | Purpose of Test | Expected Results | Actual Results | Discrepancy? | Comment |
| 1 | To check if you can add new exercise | New exercise has been added to the list.txt file | New exercise has been added to the list.txt file | No |  |
| 2 | To check if new exercise shows in the dropdown | New exercise shows in the dropdown | New exercise doesn’t show in the dropdown | Yes | New method (def update\_list) has to be created that refreshes the dropdown |
| 2.1 | To check if new exercise shows in the dropdown | New exercise shows in the dropdown | New exercise shows in the dropdown | No |  |
| 3 | To check if correct exercise is written to a file when you make a selection in dropdown | Correct exercise is written to a file when you make a selection in dropdown | Correct exercise is written to a file when you make a selection in dropdown | No |  |
| 4 | To check if correct number of reps is written to a file when you make a selection in dropdown | Correct number of reps is written to a file when you make a selection in dropdown | Correct number of reps is written to a file when you make a selection in dropdown | No |  |
| 5 | To check if correct weight is written to a file when you make a selection in dropdown | Correct weight is written to a file when you make a selection in dropdown | Correct weight is written to a file when you make a selection in dropdown | No |  |
| 6 | To check if correct date is written to a file when creating a new workout | Correct date is written to a file when creating a new workout | Correct date is written to a file when creating a new workout | No |  |
| 7 | To check if number of sets increases when you enter more sets in the same exercise | Number of sets increases when you enter more sets in the same exercise | Number of sets increases when you enter more sets in the same exercise | No |  |
| 8 | To check if number of sets defaults to 1 when you enter set for different exercise | Number of sets defaults to 1 when you enter set for different exercise | Number of sets didn’t defaults to 1 when you enter set for different exercise | Yes | Each exercise has to have its own variable for storing set amount |
| 8.1 | To check if number of sets defaults to 1 when you enter set for different exercise | Number of sets defaults to 1 when you enter set for different exercise | Number of sets defaults to 1 when you enter set for different exercise | No |  |
| 9 | To check if recommended training displays correct exercise name | Recommended training displays correct exercise name | Recommended training displays correct exercise name | No |  |
| 10 | To check if recommended training displays correct reps and weight values when reps are less than 3 | Recommended training displays more reps and lower weight when reps are less than 3 | Recommended training displays more reps and lower weight when reps are less than 3 | No |  |
| 11 | To check if recommended training displays correct reps and weight values when reps are less than 3 and weight is more than 60 | Recommended training displays more reps and lower weight (by 5kg) when reps are less than 3 and weight is more than 60 | Recommended training displays more reps and lower weight (by 5kg) when reps are less than 3 and weight is more than 60 | No |  |
| 12 | To check if recommended training displays correct reps and weight values when reps are more than 6 | Recommended training displays more reps and same weight when reps are less than 3 | Recommended training displays more reps and same weight when reps are less than 3 | No |  |
| 13 | To check if recommended training displays correct reps and weight values when reps are more 9 and weight is less than 60 | Recommended training displays less reps and more weight when reps are more than 9 and weight is less than 60 | Recommended training displays less reps and more weight when reps are more than 9 and weight is less than 60 | No |  |

# 4. Critique

The program works as intended and allows the user to track their gym workout. Every time the user opens the program, it writes a date in the file so the user can see the exact date of each workout entered in the past. User has also an option to add new exercise. Then the user can enter how many reps (repetitions) and sets they did and when they click “Add your set” button, their workout will be saved to the file and showed to them in the text box under the button. User can add multiple sets for one exercise and the program will provide information about the set number e.g. Set 1…, Set 2…, etc. When exercise is changed, the set number starts from 1.

User has also function to see the recommended training. Calculation of the recommend is quite complicated and takes into consideration a lot factors. To perform the calculation, program needs data of previous exercise that user entered (reps and weight). Then based on that it decides which calculation to run. At the first stage, it performs simple logic to determine new training e.g. if reps are less than 6, increase rep count. If rep count is more than 12 increase weight and decrease rep count, but that is not enough for a good recommendation. Each workout should have an increased volume of total weight lifted. This means that the recommended training should consider the volume as well. Volume is calculated by multiplying reps by weight e.g. 10 reps times 100kg equals to 1000 volume. The program will find the best values for the next training until the suggested volume is higher than the previous volume. In this example, the program suggests increasing weight by 5kg and keeping 10 reps (which equals to 1050 volume).

Unfortunately, the program considers only the last entered set to perform the calculation. This means that it doesn’t consider the whole volume of an exercise but only last set. This could be improved in the future so the program calculates the volume for all sets and gives a recommended training to the user with a similar amount of sets, but with increased volume in the end.

GUI is simple is this program. It doesn’t use interesting colours and the layout isn’t impressive. This is something that can be improved. Adding a more modern layout will make this program more pleasing to the eye e.g. by making it in the dark theme, and because of that, it can attract more users in the future.