Scheduling Tool Documentation

The objective of this file is to culminate all relevant information related to the usage and understanding of the Scheduling Tool software. Because this application was built for the UVA Blacksmithing club, I have not spent time to improve the user interface. As a result, this documentation will provide a helpful tutorial for first time users, as the software does not have exception handling and other features which might otherwise help users troubleshoot their issues. It is my goal to accurately and approachably explain this process so that anyone-no matter their background-can successfully use the program to create a club schedule.

Author: Dominick DeCanio

Table of Contents

Overview

Features

Background

As a result of high club interest, it became necessary for the UVA Blacksmithing Club expand the number of weekly sessions available to its members to satisfy demand. Before this change, forging sessions were limited to Friday 1-5pm each week, and anyone not available for this time was simply not able to participate in the sessions.

The Scheduling Tool was built to create weekly rosters (of 2 or 3 forging sessions per week) for the UVA Blacksmithing Club that evenly distribute forging opportunities among club members for a given semester. It is important to note, though, that it makes certain assumptions about the layout and priorities of the schedule.

Assumptions:

1. There is only room for 6 club members during each forging session
2. Because this is a college club, there are a limited number of session times that are possible for both club members and the forge.
    a. These times are: Monday through Friday from 1-5pm, with an additional session time available from 9am-1pm on Friday. Friday morning sessions are denoted "FriM," or, "friM," and Friday afternoon sessions are denoted "FriA," or, "friA," respectively.

Priorities:

1. Every club member is assigned 1 session
2. For session times where 6 or fewer members are available, those available members are automatically scheduled for this time.
3. Every club member will be scheduled for biweekly sessions if they are not already
4. For a given session that is not yet filled, for which more than 6 club members are available; club members will be scheduled for these sessions based on the total number of sessions they are already scheduled for. This ensures that club members have equal access to weekly forging sessions.

Penalty Term

The system of scheduling described thus far relies on the club members providing enough availability to forge on multiple days of the week. Without that, the club cannot increase the number of forging sessions available, and the club gets gridlocked into dividing the one session per week among 30 or more club members.

Club members who are available for less common times are, sadly, penalized for it by being automatically scheduled for all those sessions. The implication of this system is that club members who are available on Monday and Friday afternoon will have most of their forging sessions on Monday, which is the less desirable time of the two. Because of this, it is unfair to these club members if other members do not provide their full availability in order to avoid being scheduled for a less desirable time.

The penalty term is a feature of the Scheduling Tool which allows the user to give a club member a reduced number of scheduled sessions. This is the only exception to the priorities listed in the prior section. It will schedule such a member for every third week of the semester. This penalty is intended to be used very sparingly, if at all, and its main function is to enable actual repercussions if club members are unwilling to cooperate-to the detriment of the club.

Supported Schedule Layouts

The introduction of multiple weekly sessions allows for a wide variety of schedule designs. Even with 1 weekly session, it would be possible to alternate the session time between the two times most available to club members. Given 2 sessions per week, this proposition becomes more complex. I have included all possible session layouts within this program, so it is possible to create the optimal schedule for your semester's availability. I have also included an option to create a schedule with 3 sessions per week to account for further club growth.

Below is a diagram illustrating all possible schedule layouts that can be created using this software.

2 Sessions per week

0 sessions alternating

Week	Sun	Mon	Tues	Wed	Thur	FriM	FriA	Sat
a						✗	✗	
b						✗	✗	

1 session alternating

Week	Sun	Mon	Tues	Wed	Thur	FriM	FriA	Sat
a		✗					✗	
b			✗				✗	

2 sessions alternating

Week	Sun	Mon	Tues	Wed	Thur	FriM	FriA	Sat
a		✗				✗		
b			✗				✗	

3 Sessions per week

0 sessions alternating

Week	Sun	Mon	Tues	Wed	Thur	FriM	FriA	Sat
a			✗			✗	✗	
b			✗			✗	✗	

1 session alternating

Week	Sun	Mon	Tues	Wed	Thur	FriM	FriA	Sat
a		✗				✗	✗	
b			✗			✗	✗	

2 sessions alternating

Week	Sun	Mon	Tues	Wed	Thur	FriM	FriA	Sat
a		✗		✗			✗	
b			✗		✗		✗	

3 sessions alternating

Week	Sun	Mon	Tues	Wed	Thur	FriM	FriA	Sat
a		✗		✗		✗		
b			✗		✗		✗	

Output

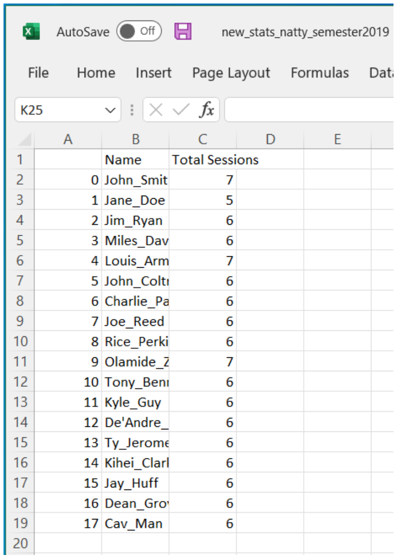
Two files are created from the Scheduling Tool software.

One is a .csv file which contains only the club schedule. This file is meant to be distributed using the club listserv, and in conjunction with the document "Personal Schedule How-To," so that club members can see their individual schedules for forging sessions.

The other file is also a .csv, but it contains summary statistics for the club schedule. This file allows the user to see how evenly the sessions are distributed and the number of sessions scheduled for each club member. This information could be useful if saved over time, as it would allow future club leaders to see the number of sessions club members get per semester, on average. With a couple excel formulas you can make a histogram of the results.

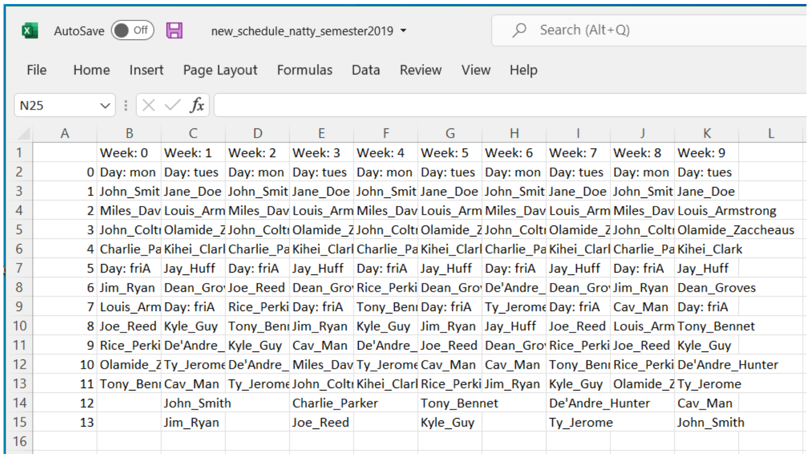
Example output:

Summary Statistics



	Name	Total Sessions
0	John_Smit	7
1	Jane_Doe	5
2	Jim_Ryan	6
3	Miles_Dav	6
4	Louis_Arm	7
5	John_Colt	6
6	Charlie_Pa	6
7	Joe_Reed	6
8	Rice_Perki	6
9	Olamide_Z	7
10	Tony_Ben	6
11	Kyle_Guy	6
12	De'Andre_	6
13	Ty_Jerome	6
14	Kihei_Clar	6
15	Jay_Huff	6
16	Dean_Gro	6
17	Cav_Man	6

Club Schedule



	Week: 0	Week: 1	Week: 2	Week: 3	Week: 4	Week: 5	Week: 6	Week: 7	Week: 8	Week: 9
0	Day: mon	Day: tues	Day: mon	Day: tues	Day: mon	Day: tues	Day: mon	Day: tues	Day: mon	Day: tues
1	John_Smit	Jane_Doe	John_Smit	Jane_Doe	John_Smit	Jane_Doe	John_Smit	Jane_Doe	John_Smit	Jane_Doe
2	Miles_Dav	Louis_Arm	Miles_Dav	Louis_Arm	Miles_Dav	Louis_Arm	Miles_Dav	Louis_Arm	Miles_Dav	Louis_Armstrong
3	John_Colt	Olamide_Z	John_Colt	Olamide_Z	John_Colt	Olamide_Z	John_Colt	Olamide_Z	John_Colt	Olamide_Zaccheaus
4	Charlie_Pa	Kihei_Clar	Charlie_Pa	Kihei_Clar	Charlie_Pa	Kihei_Clar	Charlie_Pa	Kihei_Clar	Charlie_Pa	Kihei_Clark
5	Day: friA	Jay_Huff	Day: friA	Jay_Huff	Day: friA	Jay_Huff	Day: friA	Jay_Huff	Day: friA	Jay_Huff
6	Jim_Ryan	Dean_Gro	Joe_Reed	Dean_Gro	Rice_Perki	Dean_Gro	De'Andre_	Dean_Gro	Jim_Ryan	Dean_Groves
7	Louis_Arm	Day: friA	Rice_Perki	Day: friA	Tony_Ben	Day: friA	Ty_Jerome	Day: friA	Cav_Man	Day: friA
8	Joe_Reed	Kyle_Guy	Tony_Ben	Jim_Ryan	Kyle_Guy	Jim_Ryan	Jay_Huff	Joe_Reed	Louis_Arm	Tony_Bennet
9	Rice_Perki	De'Andre_	Kyle_Guy	Cav_Man	De'Andre_	Joe_Reed	Dean_Gro	Rice_Perki	Joe_Reed	Kyle_Guy
10	Olamide_Z	Ty_Jerome	De'Andre_	Miles_Dav	Ty_Jerome	Cav_Man	Cav_Man	Tony_Ben	Rice_Perki	De'Andre_Hunter
11	Tony_Ben	Cav_Man	Ty_Jerome	John_Colt	Kihei_Clar	Rice_Perki	Jim_Ryan	Kyle_Guy	Olamide_Z	Ty_Jerome
12		John_Smith		Charlie_Parker		Tony_Bennet		De'Andre_Hunter		Cav_Man
13		Jim_Ryan		Joe_Reed		Kyle_Guy		Ty_Jerome		John_Smith

Note: There are blank spots in this schedule, indicating open forging session slots that cannot be filled using the current schduled days. Jane doe has less sessions scheduled than everyone else. These are indications that the schedule is not optimized.

Limitations & Support

The capability of this software is limited by the assumptions explained in the "Background" subsection of "Features" (Page __). Beyond the program's functional limitations, the user may experience compatibility issues with the application and their operating system. If you are experiencing compatibility issues you are welcome to visit my Github page in order to download the source code and find a solution locally.

Github link: https://github.com/domdecanio/Club-Scheduling-Tool

Preparation

Setup

There setup of the Scheduling Tool is very simple. There are two versions of the software:

- Scheduling Tool Win (Windows compatible)
- Scheduling Tool Mac (Mac OS X compatible)

Make sure to download the version that is compatible with the operating system at your disposal. If neither of these operating systems prove accessible, refer to the Github link in the “Limitation & Support” section, and install a version of python (3.8+) that is compatible with your operating system. At this point, you may run the Scheduling Tool source code directly from python.

Inputs

File Locations

The location of the main input file, and the location where you wish to save the final documents, are both required by the Scheduling Tool software. This is called a “path.”

Layout Selection

Before you begin, use the diagram in the “Supported Schedule Layouts” section to determine your desired number of:

- Sessions per week
- Sessions that will alternate

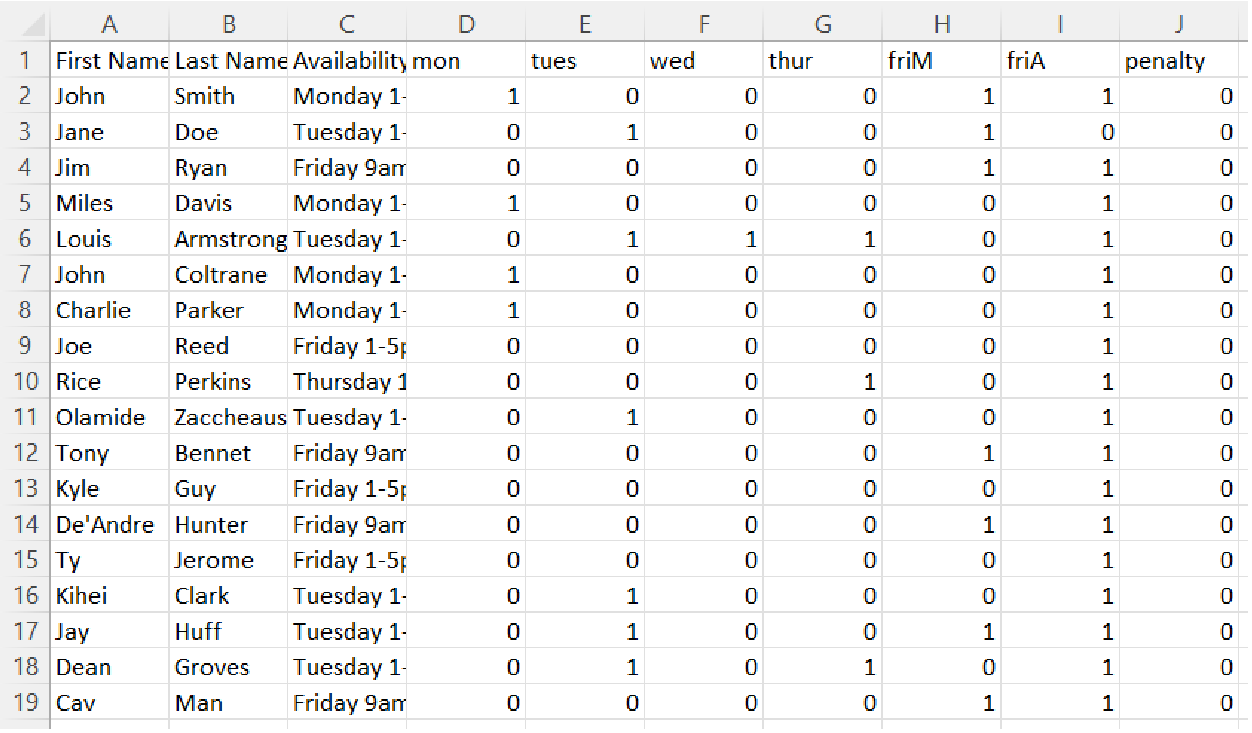
These numbers are required inputs in the Scheduling Tool software.

Member Data

In order to use the Scheduling Tool software, you must have a prepared csv file which specifies each club member, their respective availability, and their penalty term. The prepared file should mirror the snippet below, which was prepared using fake data to mimic a realistic availability pattern. The column "Availability" is not necessary; it is a hold-over from prior manipulations.

Note:

1. The file MUST be a .csv
    a. In order to save an excel file as a csv,
2. The column labels must exactly match those in the table below



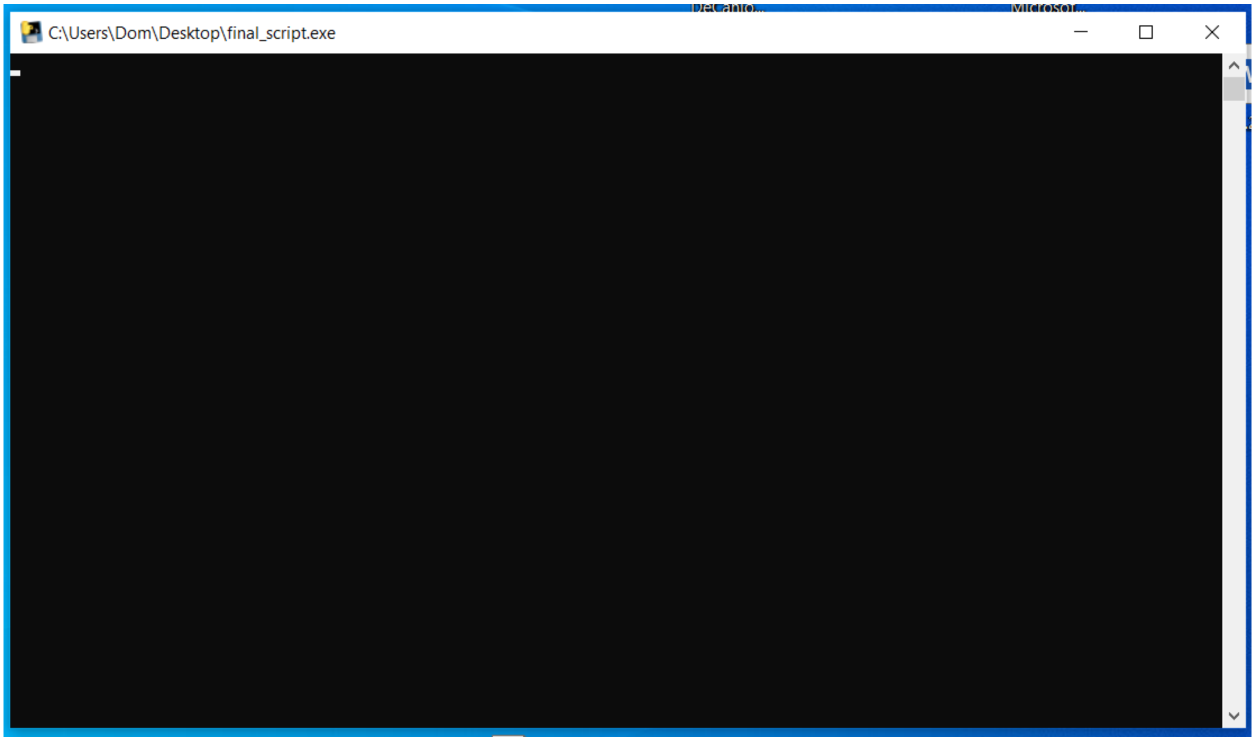
	A	B	C	D	E	F	G	H	I	J
1	First Name	Last Name	Availability	mon	tues	wed	thur	friM	friA	penalty
2	John	Smith	Monday 1-	1	0	0	0	1	1	0
3	Jane	Doe	Tuesday 1-	0	1	0	0	1	0	0
4	Jim	Ryan	Friday 9am	0	0	0	0	1	1	0
5	Miles	Davis	Monday 1-	1	0	0	0	0	1	0
6	Louis	Armstrong	Tuesday 1-	0	1	1	1	0	1	0
7	John	Coltrane	Monday 1-	1	0	0	0	0	1	0
8	Charlie	Parker	Monday 1-	1	0	0	0	0	1	0
9	Joe	Reed	Friday 1-5p	0	0	0	0	0	1	0
10	Rice	Perkins	Thursday 1	0	0	0	1	0	1	0
11	Olamide	Zaccheaus	Tuesday 1-	0	1	0	0	0	1	0
12	Tony	Bennet	Friday 9am	0	0	0	0	1	1	0
13	Kyle	Guy	Friday 1-5p	0	0	0	0	0	1	0
14	De'Andre	Hunter	Friday 9am	0	0	0	0	1	1	0
15	Ty	Jerome	Friday 1-5p	0	0	0	0	0	1	0
16	Kihei	Clark	Tuesday 1-	0	1	0	0	0	1	0
17	Jay	Huff	Tuesday 1-	0	1	0	0	1	1	0
18	Dean	Groves	Tuesday 1-	0	1	0	1	0	1	0
19	Cav	Man	Friday 9am	0	0	0	0	1	1	0

Software Usage

Runtime

Because python is not suited for applications run through executable files, the Scheduling Tool is quite slow to run. When running the application for the first time, or after restarting your computer, the application will take longer to run. I am making a particular note of this because the executable can take up to 10 minutes to run at the extreme. Once it has run once, it will take around 1 minute to 30 seconds to run subsequently.

What do I mean by “to run?” The code I wrote for this software must be compiled and accessed by the packaging software that turned my code into an executable file. While this is happening, a window will appear with a blinking cursor, an example of which is shown below.



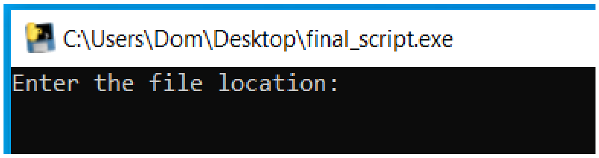
Once you see text in the window, the wait is over, and the application runs very quickly whilst building the schedule.

User Interface

The user interface for this software is very basic. This section will guide the user in properly inputting required data in order to run the Scheduling Tool successfully.

1. Enter the file location:

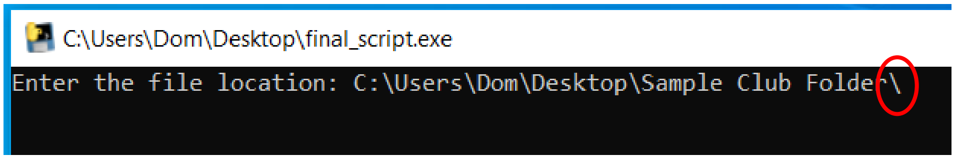
In this field, copy and paste the path of the the .csv file that contains the relevant member data. The formatting of this .csv file is covered in depth on page 8 of this document.



```
Enter the file location:
```

Once you do so, you must add a backslash to the end of the path, as this backslash is not included in the path copied from the file's properties.

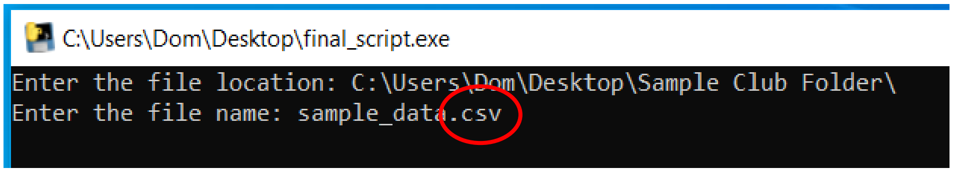
This is the correct format:



```
Enter the file location: C:\Users\Dom\Desktop\Sample Club Folder\
```

2. Enter the file name:

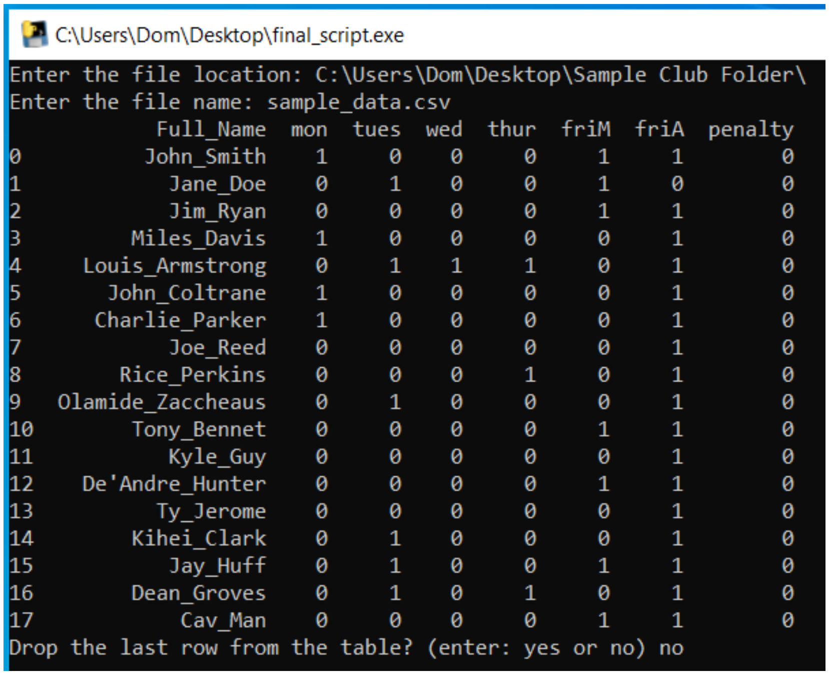
This command is looking for the name of the .csv file tha contains relevant member data. Be sure to include the format suffix in your input!



```
Enter the file location: C:\Users\Dom\Desktop\Sample Club Folder\
Enter the file name: sample_data.csv
```

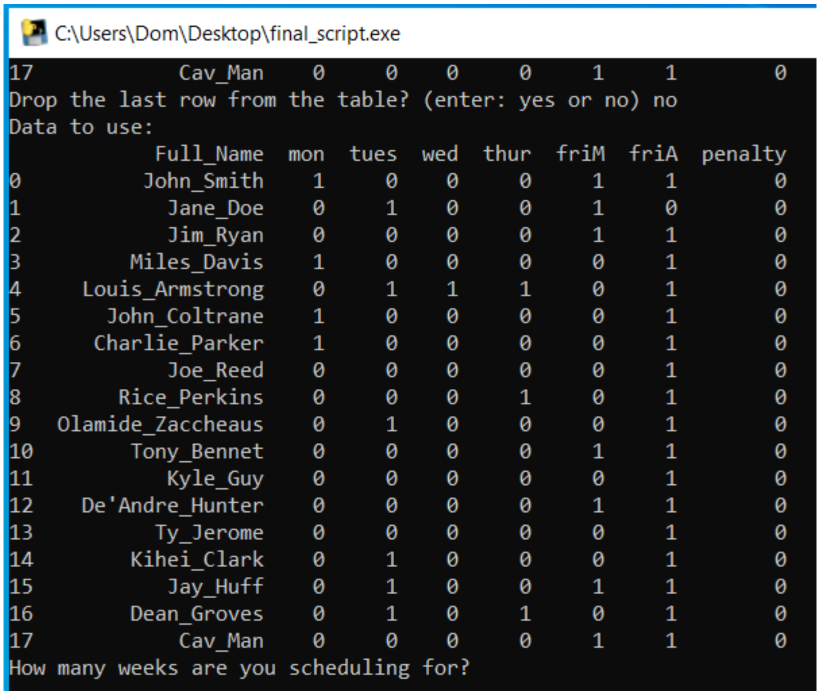
3. Drop the last row?

Preceding this input box is an output of the table as the software reads it. This allows the user to verify that the correct information has been accessed. In some instances, the last row of the data will be filled with NaN values. In this case, the user will enter “yes” in the input field. Any other input will result in the last row being retained.



```
Enter the file location: C:\Users\Dom\Desktop\Sample Club Folder\
Enter the file name: sample_data.csv
            Full_Name  mon  tues  wed  thur  friM  friA  penalty
0          John_Smith    1     0    0     0     1     1        0
1            Jane_Doe    0     1    0     0     1     0        0
2            Jim_Ryan    0     0    0     0     1     1        0
3         Miles_Davis    1     0    0     0     0     1        0
4     Louis_Armstrong    0     1    1     1     0     1        0
5       John_Coltrane    1     0    0     0     0     1        0
6      Charlie_Parker    1     0    0     0     0     1        0
7            Joe_Reed    0     0    0     0     0     1        0
8        Rice_Perkins    0     0    0     1     0     1        0
9   Olamide_Zaccheaus    0     1    0     0     0     1        0
10        Tony_Bennet    0     0    0     0     1     1        0
11           Kyle_Guy    0     0    0     0     0     1        0
12    De'Andre_Hunter    0     0    0     0     1     1        0
13          Ty_Jerome    0     0    0     0     0     1        0
14        Kihei_Clark    0     1    0     0     0     1        0
15           Jay_Huff    0     1    0     0     1     1        0
16        Dean_Groves    0     1    0     1     0     1        0
17            Cav_Man    0     0    0     0     1     1        0
Drop the last row from the table? (enter: yes or no) no
```

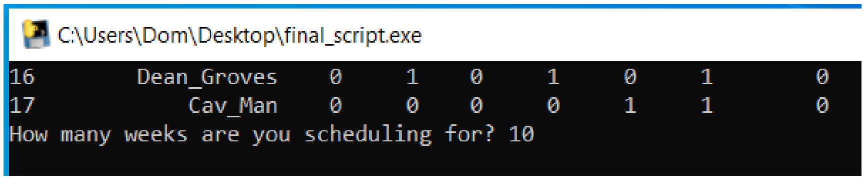
Following this input, the software will output the data again to allow the user to again verify that the data to be used in the software is correct.



```
17            Cav_Man    0     0    0     0     1     1        0
Drop the last row from the table? (enter: yes or no) no
Data to use:
            Full_Name  mon  tues  wed  thur  friM  friA  penalty
0          John_Smith    1     0    0     0     1     1        0
1            Jane_Doe    0     1    0     0     1     0        0
2            Jim_Ryan    0     0    0     0     1     1        0
3         Miles_Davis    1     0    0     0     0     1        0
4     Louis_Armstrong    0     1    1     1     0     1        0
5       John_Coltrane    1     0    0     0     0     1        0
6      Charlie_Parker    1     0    0     0     0     1        0
7            Joe_Reed    0     0    0     0     0     1        0
8        Rice_Perkins    0     0    0     1     0     1        0
9   Olamide_Zaccheaus    0     1    0     0     0     1        0
10        Tony_Bennet    0     0    0     0     1     1        0
11           Kyle_Guy    0     0    0     0     0     1        0
12    De'Andre_Hunter    0     0    0     0     1     1        0
13          Ty_Jerome    0     0    0     0     0     1        0
14        Kihei_Clark    0     1    0     0     0     1        0
15           Jay_Huff    0     1    0     0     1     1        0
16        Dean_Groves    0     1    0     1     0     1        0
17            Cav_Man    0     0    0     0     1     1        0
How many weeks are you scheduling for?
```

4. How many weeks?

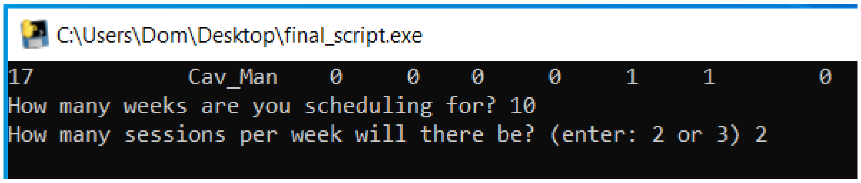
Simply input the total number of weeks the desired schedule should include. Note that you must account for spring and fall break, and forging sessions ususally end on the last week of classes.



```
16        Dean_Groves    0     1    0     1     0     1        0
17            Cav_Man    0     0    0     0     1     1        0
How many weeks are you scheduling for? 10
```

5. How many sessions?

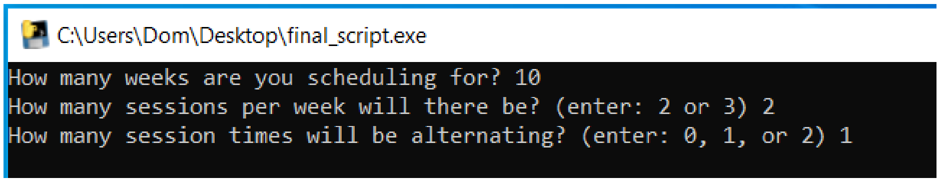
Input the number of forging sessions that will occur during each scheduled week.



```
17            Cav_Man    0     0    0     0     1     1        0
How many weeks are you scheduling for? 10
How many sessions per week will there be? (enter: 2 or 3) 2
```

6. How many alternating?

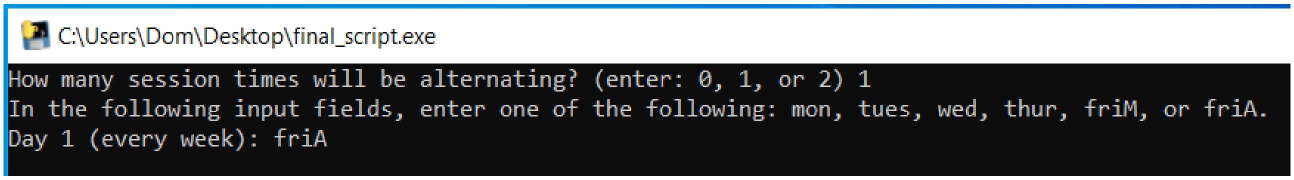
Input the number of forging sessions whose scheduled time is not the same each week.



```
How many weeks are you scheduling for? 10
How many sessions per week will there be? (enter: 2 or 3) 2
How many session times will be alternating? (enter: 0, 1, or 2) 1
```

7. Day x (every week):

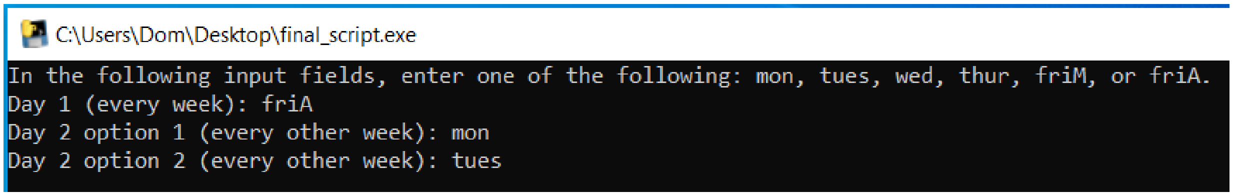
This field will occur if the number of alternating session times is less than the number of weekly sessions. The entered session time, in this case Friday afternoon (1-5pm), will be on the schedule every week.



```
How many session times will be alternating? (enter: 0, 1, or 2) 1
In the following input fields, enter one of the following: mon, tues, wed, thur, friM, or friA.
Day 1 (every week): friA
```

8. Day x (every other week):

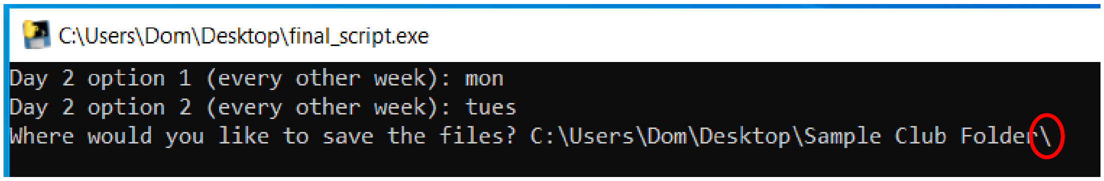
This field will occur if the number of alternating session times is greater than zero. This input is structured to follow the club resources, not the session time. Think of this as alternating the session time around for the "Day" the club has funding for. Thus, "option 1," and, "option 2," are the times that will alternate with one another for a given two week cycle. The example below exactly mirrors the diagram for "2 sessions per week, 1 session alternating," on page 4 of this document.



```
In the following input fields, enter one of the following: mon, tues, wed, thur, friM, or friA.
Day 1 (every week): friA
Day 2 option 1 (every other week): mon
Day 2 option 2 (every other week): tues
```

9. Where to save files?

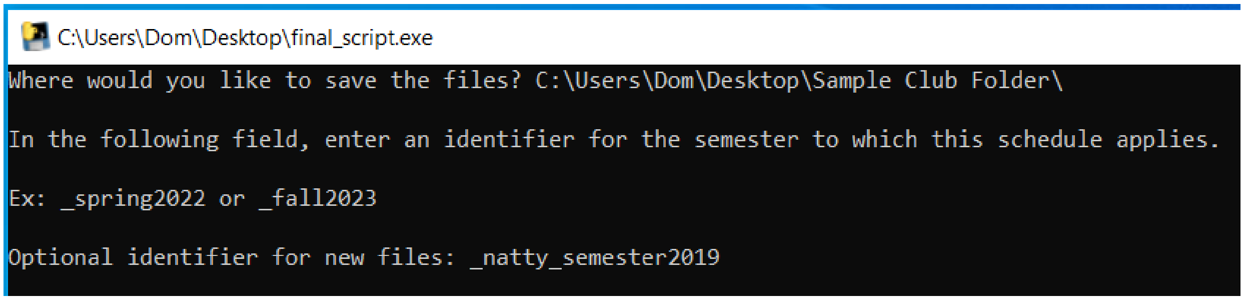
This input field is asking for the location where the user wants to have the output files created. To place them in the same folder as the input .csv file, simply paste the same path into this field. Remember to add the backslash just as in step 1!



```
Day 2 option 1 (every other week): mon
Day 2 option 2 (every other week): tues
Where would you like to save the files? C:\Users\Dom\Desktop\Sample Club Folder\
```

10. Optional Identifier:

This field is intended to help users organize and meaningfully label club schedules. The inputted string will follow both the base names "new_stats" and "new_schedule" for the schedule statistics and schedule outputs respectively.



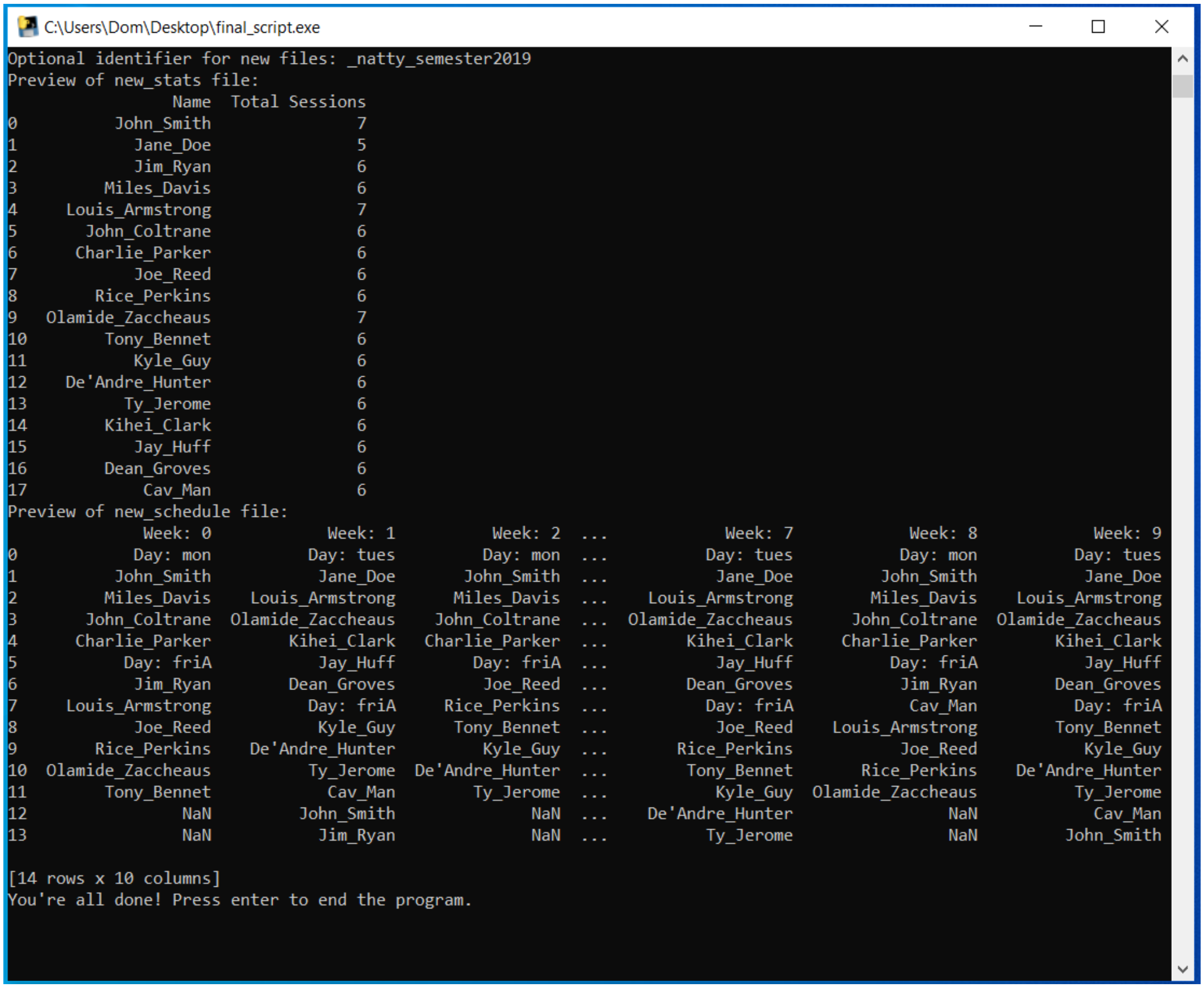
```
Where would you like to save the files? C:\Users\Dom\Desktop\Sample Club Folder\

In the following field, enter an identifier for the semester to which this schedule applies.

Ex: _spring2022 or _fall2023

Optional identifier for new files: _natty_semester2019
```

11. Preview & Completed

You’re done!

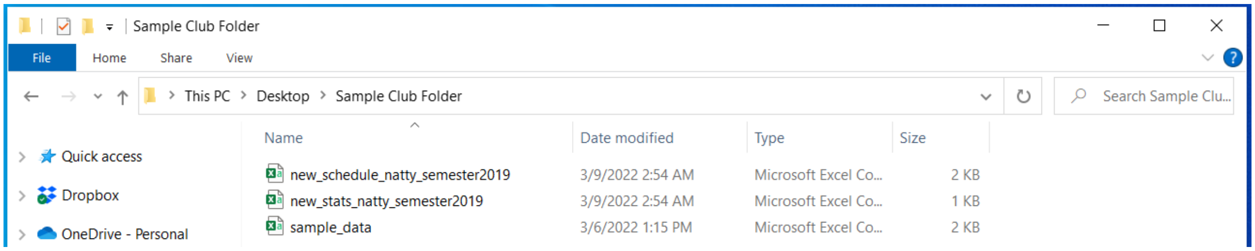
The software creates a preview of the two output files, and the user is free to press enter to close the program whenever they wish.

Example outputs are shown on page 5 of this document. Note that those outputs correspond to the inputs demonstrated in this section.



```
Optional identifier for new files: _natty_semester2019
Preview of new_stats file:
                 Name  Total Sessions
0          John_Smith               7
1            Jane_Doe               5
2            Jim_Ryan               6
3         Miles_Davis               6
4     Louis_Armstrong               7
5       John_Coltrane               6
6      Charlie_Parker               6
7            Joe_Reed               6
8        Rice_Perkins               6
9   Olamide_Zaccheaus               7
10        Tony_Bennet               6
11           Kyle_Guy               6
12    De'Andre_Hunter               6
13          Ty_Jerome               6
14        Kihei_Clark               6
15           Jay_Huff               6
16        Dean_Groves               6
17            Cav_Man               6
Preview of new_schedule file:
              Week: 0            Week: 1          Week: 2  ...            Week: 7            Week: 8            Week: 9
0            Day: mon          Day: tues         Day: mon  ...          Day: tues           Day: mon          Day: tues
1          John_Smith           Jane_Doe       John_Smith  ...           Jane_Doe         John_Smith           Jane_Doe
2         Miles_Davis    Louis_Armstrong      Miles_Davis  ...    Louis_Armstrong        Miles_Davis    Louis_Armstrong
3       John_Coltrane  Olamide_Zaccheaus    John_Coltrane  ...  Olamide_Zaccheaus      John_Coltrane  Olamide_Zaccheaus
4      Charlie_Parker        Kihei_Clark   Charlie_Parker  ...        Kihei_Clark     Charlie_Parker        Kihei_Clark
5          Day: friA           Jay_Huff        Day: friA  ...           Jay_Huff          Day: friA           Jay_Huff
6            Jim_Ryan        Dean_Groves         Joe_Reed  ...        Dean_Groves           Jim_Ryan        Dean_Groves
7     Louis_Armstrong          Day: friA     Rice_Perkins  ...          Day: friA            Cav_Man          Day: friA
8            Joe_Reed           Kyle_Guy      Tony_Bennet  ...           Joe_Reed    Louis_Armstrong        Tony_Bennet
9        Rice_Perkins    De'Andre_Hunter         Kyle_Guy  ...       Rice_Perkins           Joe_Reed           Kyle_Guy
10  Olamide_Zaccheaus          Ty_Jerome  De'Andre_Hunter  ...        Tony_Bennet       Rice_Perkins    De'Andre_Hunter
11        Tony_Bennet            Cav_Man        Ty_Jerome  ...           Kyle_Guy  Olamide_Zaccheaus          Ty_Jerome
12                NaN         John_Smith              NaN  ...    De'Andre_Hunter                NaN            Cav_Man
13                NaN           Jim_Ryan              NaN  ...          Ty_Jerome                NaN         John_Smith

[14 rows x 10 columns]
You're all done! Press enter to end the program.
```

Here is a snippet of my folder “Sample Club Folder”. Both of the output files were saved in the same location as the original members data .csv file.



Name	Date modified	Type	Size
new_schedule_natty_semester2019	3/9/2022 2:54 AM	Microsoft Excel Co...	2 KB
new_stats_natty_semester2019	3/9/2022 2:54 AM	Microsoft Excel Co...	1 KB
sample_data	3/6/2022 1:15 PM	Microsoft Excel Co...	2 KB

Troubleshooting

If there is any error in your inputs, the software will error out and close immediately upon the user entering that data. This will help you find your mistakes in inputs.

Things to look for:

- Typos
    - This is by far the largest cause of errors. It is very easy to forget an underscore or a capitalized letter, and small errors like these will cause the Scheduling Tool software to error.
- Slash orientation
    - If you have checked all your input thoroughly for typographical errors, and you are running into trouble while loading your members data .csv into the software or outputting the files from the software, there is one more technique you can try. Copy the desired path as usual but change all the backslashes (\) to forward slashes (/), subsequently entering this new path into the software. This is a frequent culprit of errors.