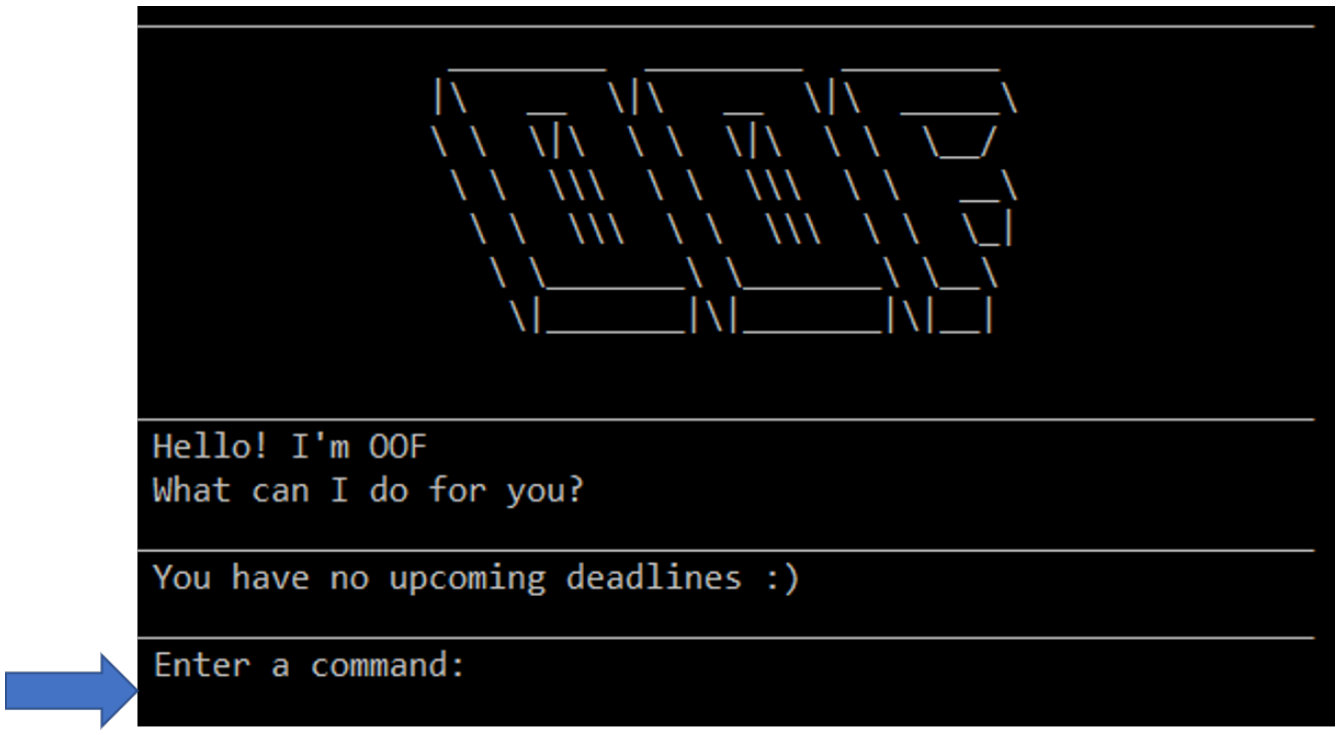
OOF (Outstanding Organisation Friend) - User Guide

By: `Team W17-4` Since: `Aug 2019` Licence: `MIT`

1. Introduction



```
 _________  _________  _________
|\   __  \|\   __  \|\  _____\
\ \  \|\  \ \  \|\  \ \  \__/
 \ \  \\\  \ \  \\\  \ \   __\
  \ \  \\\  \ \  \\\  \ \  \_|
   \ \_______\ \_______\ \__\
    \|_______|\|_______|\|__|

Hello! I'm OOF
What can I do for you?

You have no upcoming deadlines :)

Enter a command:
```

Figure 1. OOF welcome screen

1.1. What is OOF?

OOF (Oustanding Organisational Friend) is a Command Line Interface (CLI) program that allows you to save your tasks, assignments, modules taken, etc. **OOF** is catered towards university students who want to use a desktop application to manage their tasks in a friendly and efficient manner. **OOF** is optimized for users who prefer to work with the CLI.

1.2. What can OOF do?

Besides saving your tasks very effectively in persistent storage, **OOF** allows your tasks to be displayed in friendly formats such as calendar format or a tabular format where your tasks are sorted chronologically for any particular week. You can also allow **OOF** to remind you of tasks that are expiring based on a customisable threshold. On top of that, you can track your progress and see if you are on track by using our tracking feature.

1.3. How does OOF address our target audience?

Most university students are often busy and **OOF** aims to reduce the time students spend on managing their tasks. **OOF** allows students to enter one-liner commands quickly into our program and hence spend less time logging down the tasks to be done. Furthermore, **OOF** allows tasks to be viewed in insightful formats and also provides timely reminders for tasks with their deadlines nearing.

1.4. What is this guide for?

This guide aims to educate you on how to use our application by providing example usages of all its features. The features can be found in Section 3, “Features” section.

Interested in using **OOF** to plan your timetable more effectively? Jump to Section 2, "Quick Start" to get started! Enjoy!

2. Quick Start

2.1. Setting up

1. Ensure you have Java 11 or above installed on your computer.
   a. For `windows` users:
      i. Download the latest release here.
      ii. Open a `console` window.
      iii. Navigate to the directory containing our jar file.
      iv. Press `ALT` + `ENTER` to enter full-screen mode.

         If the output is large, windows will wrap the output by default. This makes the output unreadable and hence you should enter full-screen mode to prevent that from happening.

      v. Run the command "java -jar ***[X]***.jar". The application should load within a few seconds.

         [X] refers to the name of our latest jar release.

   b. For `mac` users:
      i. Download the latest release here.
      ii. Open a `terminal`.
      iii. Navigate to the directory containing our jar file.
      iv. Run the command `tput rmam`.

         This command disables line wrapping which is essential for our output to be sensible to you. You can undo this setting by typing the command `tput smam`. Note that there is no horizontal scrolling feature in terminal. Thus, for bigger output, you may not be able to see the full output. You can attempt to work around this limitation by using the system level feature in `OSx` by holding the `SHIFT` key and scrolling using your mouse scroll wheel.

      v. Run the command "java -jar ***[X]***.jar". The application should load within a few seconds.

         [X] refers to the name of our latest jar release.

   c. For `linux` users:
      i. Download the latest release here.

ii. Open a `terminal`.

iii. Navigate to the directory containing our jar file.

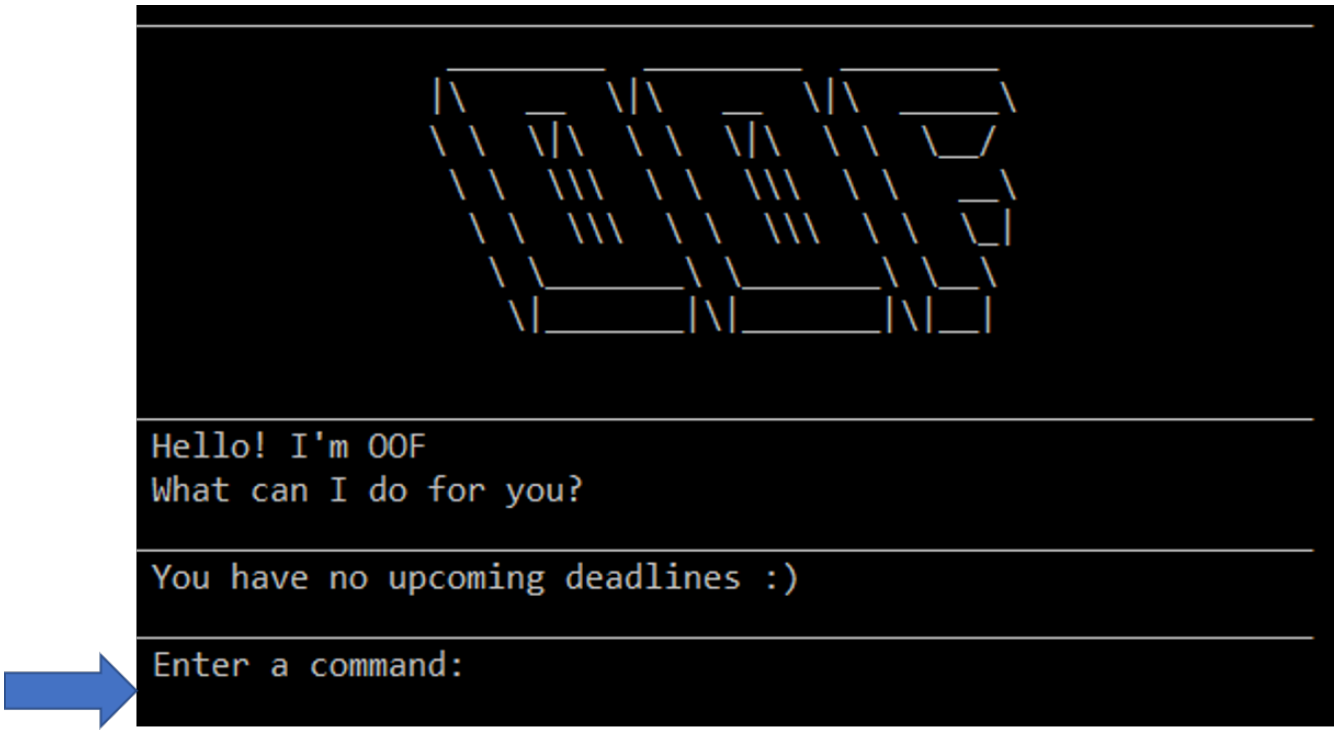
iv. Run the command `setterm -linewrap off`.

> This command disables line wrapping which is essential for our output to be sensible to you. You can undo this setting by typing the command `setterm -linewrap on`. Note that there is no horizontal scrolling feature in terminal. Thus, for bigger output, you may not be able to see the full output. You can attempt to work around this limitation by zooming out before keying in our commands. You can do so by pressing the combination `CTRL` + `-` multiple times. You can also undo this by pressing the combination `CTRL` + `SHIFT` + `=` or `CTRL` + `+`.

v. Run the command “java -jar ***[X]***.jar”. The application should load within a few seconds.

> ***[X]*** refers to the name of our latest jar release.

2.2. Sample commands



```
 ________  ________  ________
|\   __  \|\   __  \|\  _____\
\ \  \|\  \ \  \|\  \ \  \__/
 \ \  \\\  \ \  \\\  \ \   __\
  \ \  \\\  \ \  \\\  \ \  \_|
   \ \_______\ \_______\ \__\
    \|_______|\|_______|\|__|

Hello! I'm OOF
What can I do for you?

You have no upcoming deadlines :)

Enter a command:
```

Figure 2. OOF welcome screen

1. Type a task description in the terminal and press `Enter` to run it.
   e.g. typing **`help`** and pressing `Enter` will list the commands present.

2. Some example commands you can try:

   - **`deadline homework /by 12-12-2019 11:11`** : adds a task called `homework` to the saved tasks with the deadline `12-12-2019 11:11`.

   - **`calendar`** : displays all saved tasks in a calendar view.

    - `Bye` : exits the application.

A summary of all the features available in **OOF** can be found in Section 5, “Command Summary”.

Refer to Section 3, “Features” for details of each command.

3. Features

In this section, the expected command format will be introduced, and you can expect to learn the various commands you can use.

Command Format

- Words in `UPPER_CASE` are the parameters to be supplied by the user e.g. `deadline DESCRIPTION /by DD-MM-YYYY HH:MM`
- The maximum length for a task’s `description` is `20` characters.
- The maximum lengths for a `module code` / `year` and `module name` / `semester` are `10` and `100` characters respectively.

Don’t worry if you do not understand everything at once.
There are plentiful examples provided to aid your understanding of the commands' usage.

3.1. Navigation

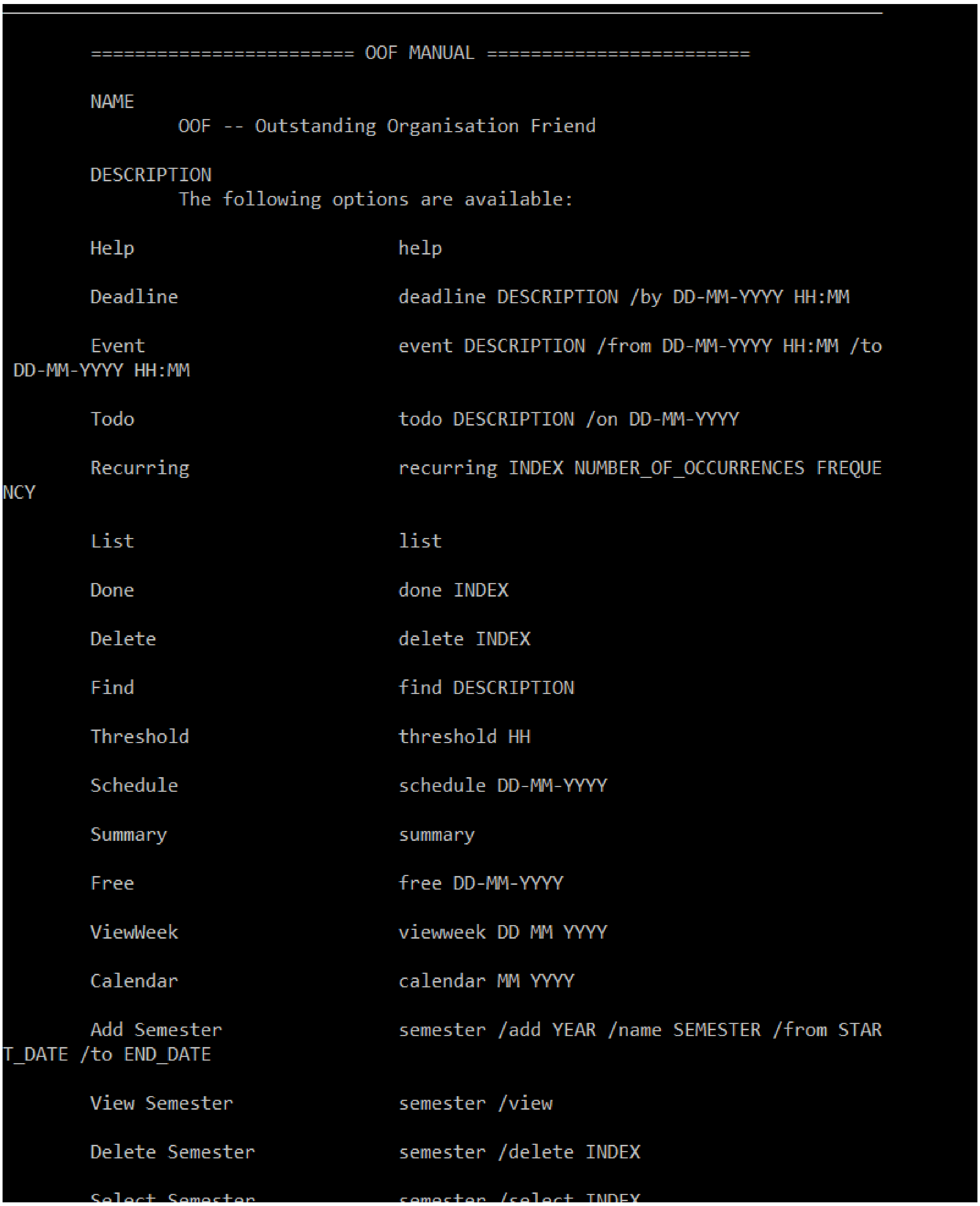
3.1.1. Viewing the manual: `help`

Shows you a list of commands that can be used.

Format: `help`

Example:

- User enters `help`



```
        ======================== OOF MANUAL ========================

        NAME
                OOF -- Outstanding Organisation Friend

        DESCRIPTION
                The following options are available:

        Help                        help

        Deadline                    deadline DESCRIPTION /by DD-MM-YYYY HH:MM

        Event                       event DESCRIPTION /from DD-MM-YYYY HH:MM /to
 DD-MM-YYYY HH:MM

        Todo                        todo DESCRIPTION /on DD-MM-YYYY

        Recurring                   recurring INDEX NUMBER_OF_OCCURRENCES FREQUE
NCY

        List                        list

        Done                        done INDEX

        Delete                      delete INDEX

        Find                        find DESCRIPTION

        Threshold                   threshold HH

        Schedule                    schedule DD-MM-YYYY

        Summary                     summary

        Free                        free DD-MM-YYYY

        ViewWeek                    viewweek DD MM YYYY

        Calendar                    calendar MM YYYY

        Add Semester                semester /add YEAR /name SEMESTER /from STAR
T_DATE /to END_DATE

        View Semester               semester /view

        Delete Semester             semester /delete INDEX

        Select Semester             semester /select INDEX
```

Figure 3. Output of Help Command

Usage of all the features is shown to you if `help` is entered.

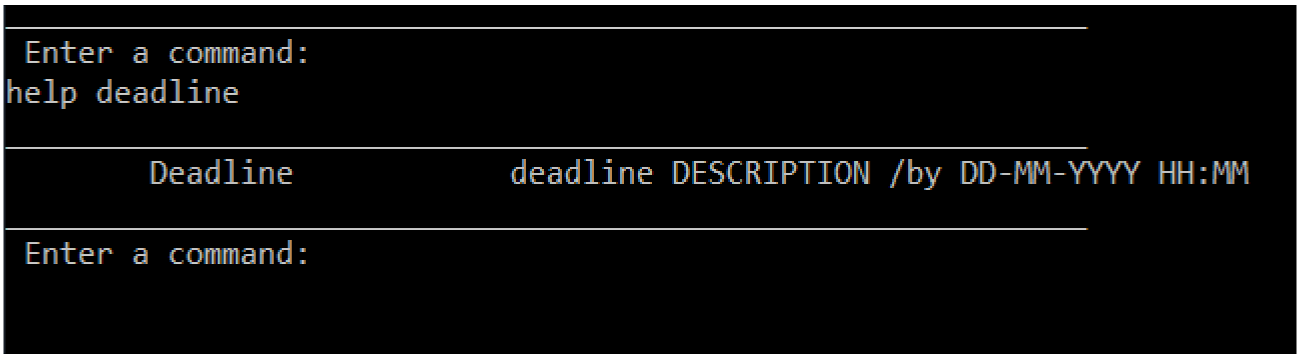
3.1.2. Viewing the usage of individual commands: `help`

Shows you the specific usage for the command you have entered.

Format: `help COMMAND`

Example:

- `help Deadline`



```
 Enter a command:
help deadline

        Deadline            deadline DESCRIPTION /by DD-MM-YYYY HH:MM

 Enter a command:
```

Figure 4. Example of help COMMAND usage

Correct syntax of adding a `deadline` is shown.

3.1.3. Exiting the program: `bye`

Exits the program.

Format: `bye`

3.2. Semesters

You can plan ahead for your entire university journey using a few simple commands.

3.2.1. Adding semesters: `semester /add`

You can add a semester to manage your modules.

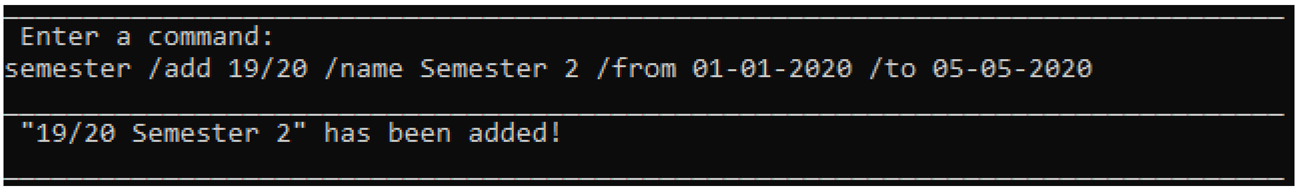
Format: `semester /add YEAR /name SEMESTER /from START_DATE /to END_DATE`

- `YEAR` represents name of the academic year, `SEMESTER` represents name of the semester, `START_DATE` and `END_DATE` represents the start and end date in `dd-MM-yyyy HH:mm` format.

`YEAR` and `SEMESTER` have a character limit of 10 and 100 characters respectively.

Example:

- `semester /add 19/20 /name Semester 2 /from 01-01-2020 /to 05-05-2020`



```
 Enter a command:
semester /add 19/20 /name Semester 2 /from 01-01-2020 /to 05-05-2020

 "19/20 Semester 2" has been added!
```

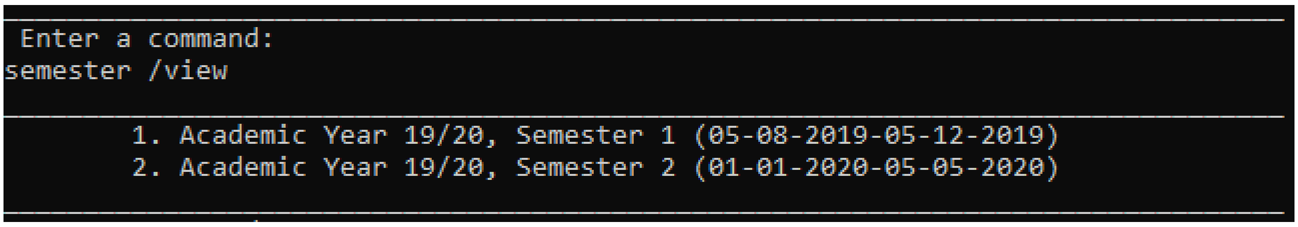
Figure 5. Adding a semester

Adds a semester for Academic Year 19/20, Semester 2 which lasts from 01-01-2020 to 05-05-2020.

3.2.2. Viewing semester data: `semester /view`

You can use this command to display all the semesters you have added.

Format: `semester /view`



```
 Enter a command:
semester /view
________________________________________________________________________________
        1. Academic Year 19/20, Semester 1 (05-08-2019-05-12-2019)
        2. Academic Year 19/20, Semester 2 (01-01-2020-05-05-2020)
________________________________________________________________________________
```

Figure 6. Viewing a semester

3.2.3. Removing semester data: `semester /delete`

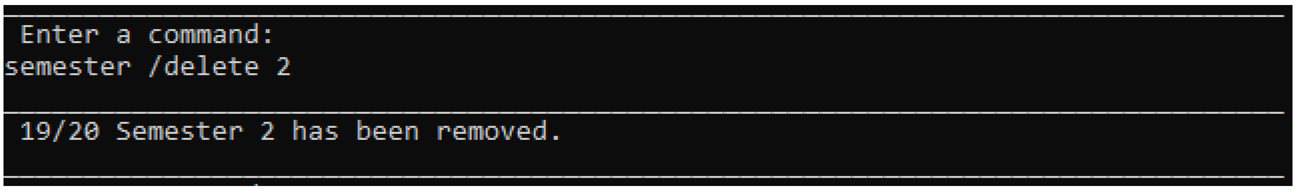
You can remove unwanted data if you have accidentally added an unwanted semester.

Format: `semester /delete INDEX`

- The `INDEX` refers to the index number displayed in the list of semesters recorded. (`semester /view` can be used to display the added semesters).

Example:

- `semester /delete 2`



```
 Enter a command:
semester /delete 2
________________________________________________________________________________
 19/20 Semester 2 has been removed.
________________________________________________________________________________
```

Figure 7. Deleting a Semester.

3.2.4. Selecting a semester: `semester /select`

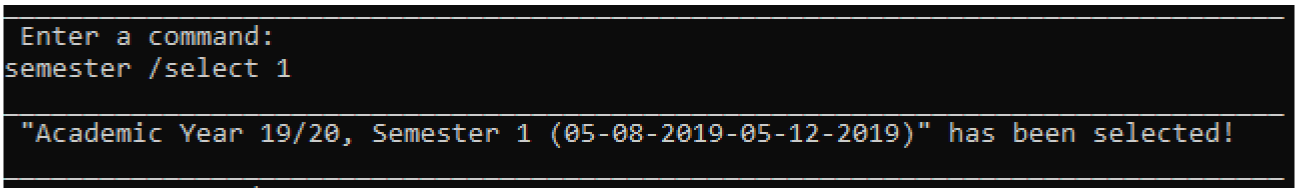
You can select a semester in order to add modules to under a semester.

Format: `semester /select INDEX`

- The `INDEX` refers to the index number displayed in the list of semesters recorded. (`semester /view` can be used to display the added semesters).

Example:

- `semester /select 1`



```
 Enter a command:
semester /select 1
________________________________________________________________________________
 "Academic Year 19/20, Semester 1 (05-08-2019-05-12-2019)" has been selected!
________________________________________________________________________________
```

Figure 8. Selecting a semester

Selects the 1st semester in the list of semesters.

3.3. Modules

You can keep track of your modules each semester with the help of the module commands.

All commands under modules require a semester to be selected using `semester /select`.

3.3.1. Adding module data: `module /add`

You can add a module into `Oof` to manage your lessons and assessments.

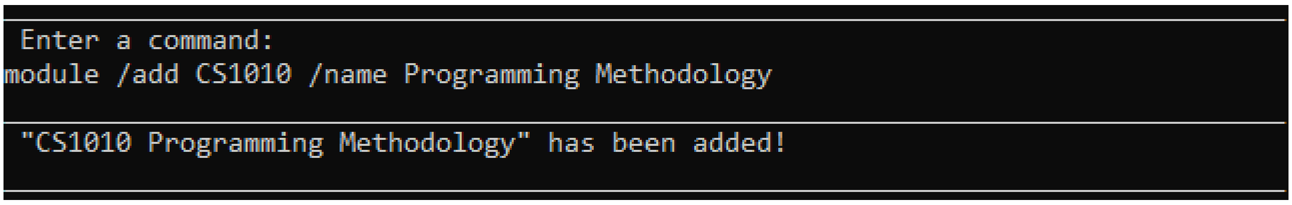
Format: `module /add MODULE_CODE /name MODULE_NAME`

- `MODULE_CODE` represents the module code and `MODULE_NAME` represents the module name.

`MODULE_CODE` and `MODULE_NAME` have a character limit of 10 and 100 characters respectively.

Example:

- `module /add CS1010 /name Programming Methodology`



```
 Enter a command:
module /add CS1010 /name Programming Methodology

 "CS1010 Programming Methodology" has been added!
```

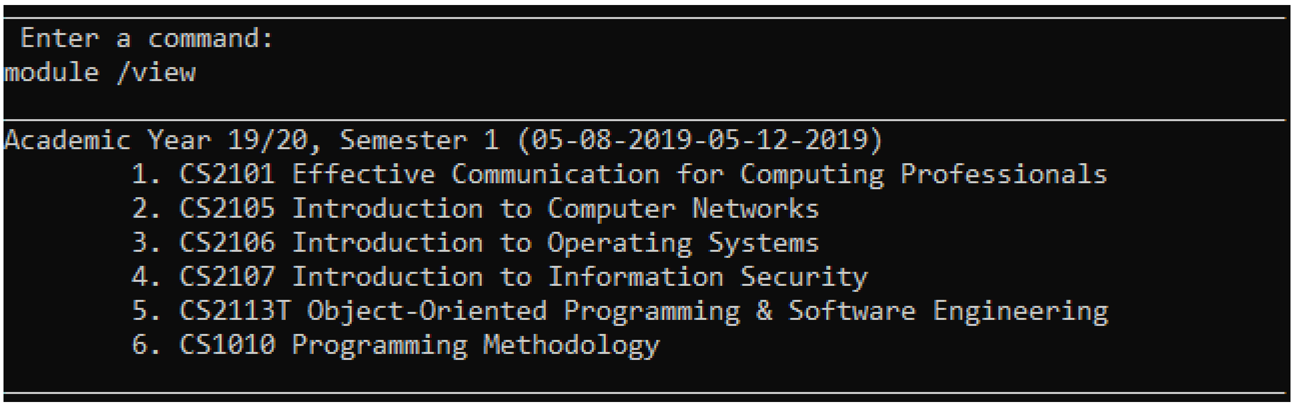
Figure 9. Adding a module

Adds a module with module code "CS1010" and name as "Programming Methodology".

3.3.2. Viewing module data: `module /view`

You can display all modules in order to have a quick overview of the modules you are taking this semester.

Format: `module /view`



```
 Enter a command:
module /view

Academic Year 19/20, Semester 1 (05-08-2019-05-12-2019)
        1. CS2101 Effective Communication for Computing Professionals
        2. CS2105 Introduction to Computer Networks
        3. CS2106 Introduction to Operating Systems
        4. CS2107 Introduction to Information Security
        5. CS2113T Object-Oriented Programming & Software Engineering
        6. CS1010 Programming Methodology
```

Figure 10. Viewing a module

3.3.3. Removing module data: `module /delete`

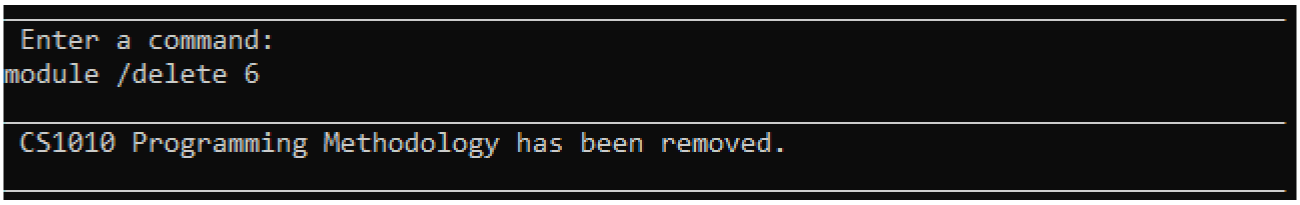
You can remove unwanted data if you have accidentally added a wrong module.

Format: `module /delete INDEX`

- The `INDEX` refers to the index number displayed in the list of modules recorded. `module /view` can be used to display the saved semesters).

Example:

- `module /delete 6`



```
 Enter a command:
module /delete 6

 CS1010 Programming Methodology has been removed.
```

Figure 11. Deleting a module

Deletes the 4th module in the list of modules.

3.3.4. Selecting a module: `module /select`

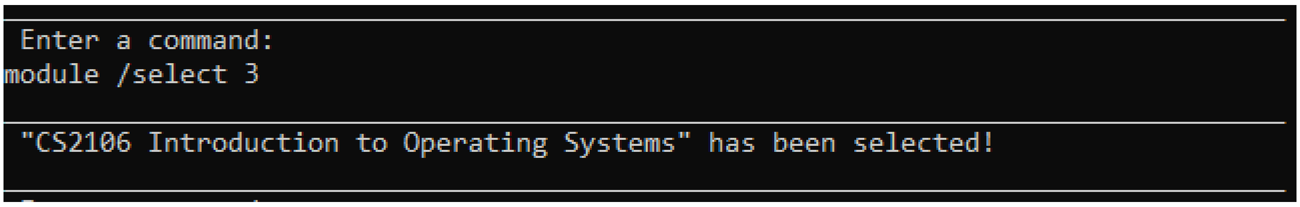
You can select a module in order to add lessons for a module.

Format: `module /select INDEX`

- The `INDEX` refers to the index number displayed in the list of modules recorded. (`module /view` can be used to display the added modules).

Example:

- `module /select 3`



```
 Enter a command:
module /select 3

 "CS2106 Introduction to Operating Systems" has been selected!
```

Figure 12. Selecting a module

Selects the 1st module in the list of modules.

3.4. Lessons

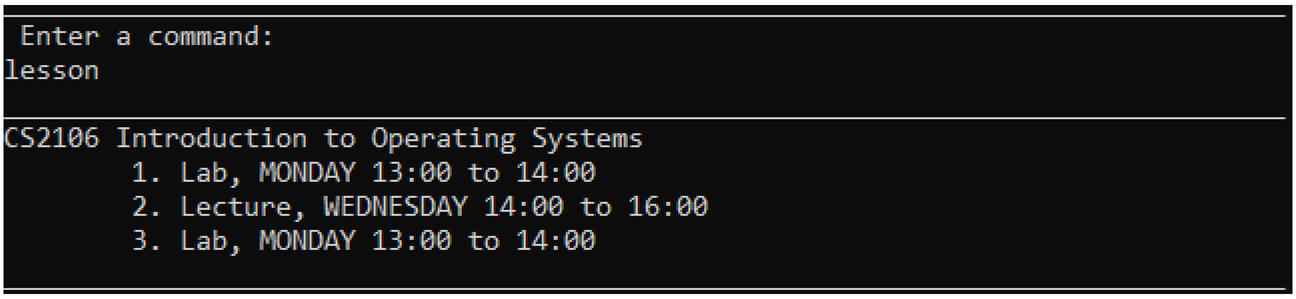
Keep track of your lessons for each module with the use of lesson commands!

All commands under lesson require a module to be selected using `module /select`.

3.4.1. Viewing lesson data: `lesson`

You can display all the lessons you have added if you wish to view all lessons for a module.

Format: `lesson`



```
 Enter a command:
lesson

CS2106 Introduction to Operating Systems
        1. Lab, MONDAY 13:00 to 14:00
        2. Lecture, WEDNESDAY 14:00 to 16:00
        3. Lab, MONDAY 13:00 to 14:00
```

Figure 13. Viewing list of modules

3.4.2. Adding lesson data: `lesson /add`

You can add a lesson into `Oof`.

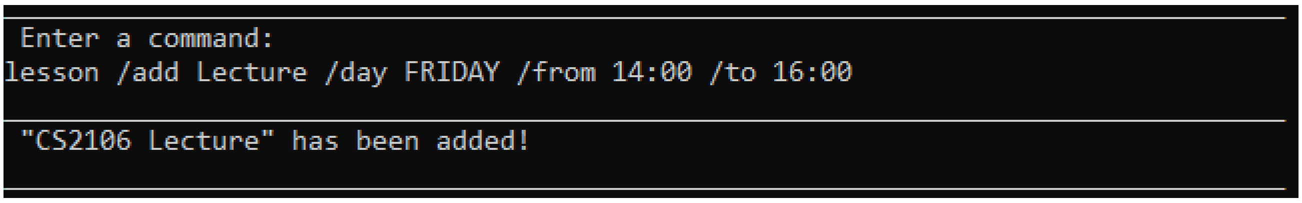
Format: `lesson /add NAME /day DAY /from START_TIME /to END_TIME`

`NAME` has a character limit of 20 characters.

- `NAME` of the lesson can have multiple words, not just limited to single-word descriptions.
- `DAY` of the lesson ranges from `MONDAY` to `SUNDAY`.
- `START_TIME` and `END_TIME` have to **strictly** be in the `HH:MM` format.

Example:

- `lesson /add lecture /day FRIDAY /from 14:00 /to 16:00`
  Adds a lecture on Friday from 14:00 to 16:00 for the selected module.



```
 Enter a command:
lesson /add Lecture /day FRIDAY /from 14:00 /to 16:00

 "CS2106 Lecture" has been added!
```

Figure 14. Adding a lesson

3.4.3. Removing lesson data: `lesson /delete`

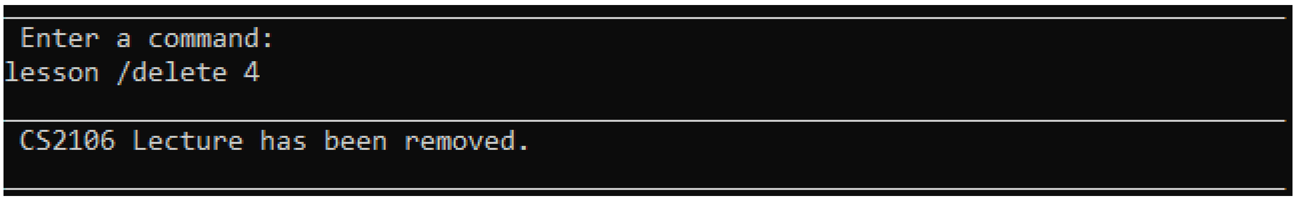
You can remove unwanted data if you have added the wrong date for a lesson.

Format: `lesson /delete INDEX`

- The `INDEX` refers to the index number displayed in the list of lessons recorded. `lesson /view` can be used to display the saved lessons).

Example:

- `lesson /delete 4`



```
 Enter a command:
lesson /delete 4

 CS2106 Lecture has been removed.
```

Figure 15. Deleting a lesson

Deletes the 4th lesson in the list of lessons.

3.5. Adding tasks

3.5.1. Adding assessment data: `assessment`

You can keep track of assessments by adding assessments.

Requires a module to be selected using `module /select`.

Format: `assessment DESCRIPTION /from DD-MM-YYYY HH:MM /to DD-MM-YYYY HH:MM`

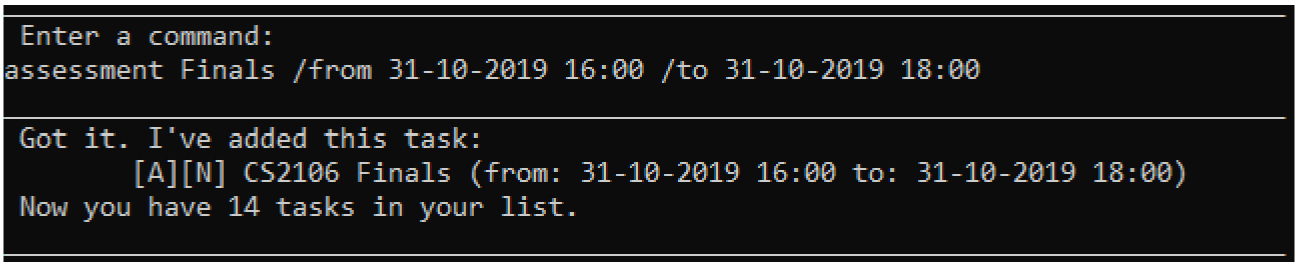
`DESCRIPTION` has a character limit of 20 characters.

- `Description` of the assessment can have multiple words, not just limited to single-word descriptions.
- `Date and time` have to **strictly** be in the format as stated above.

Example:

- `assessment Finals /from 31-10-2019 16:00 /to 31-10-2019 18:00`

Adds an assessment for current selected `Module` (`CS2106` in the example) with name, start and end time as `Finals`, `31-10-2019 13:00`, `31-10-2019 15:00` respectively.



```
 Enter a command:
assessment Finals /from 31-10-2019 16:00 /to 31-10-2019 18:00

 Got it. I've added this task:
        [A][N] CS2106 Finals (from: 31-10-2019 16:00 to: 31-10-2019 18:00)
 Now you have 14 tasks in your list.
```

Figure 16. Adding an assessment

3.5.2. Adding assignment data: `assignment`

You can use this command to keep track of an assignment for a particular module.

> Requires a module to be selected using `module /select`.

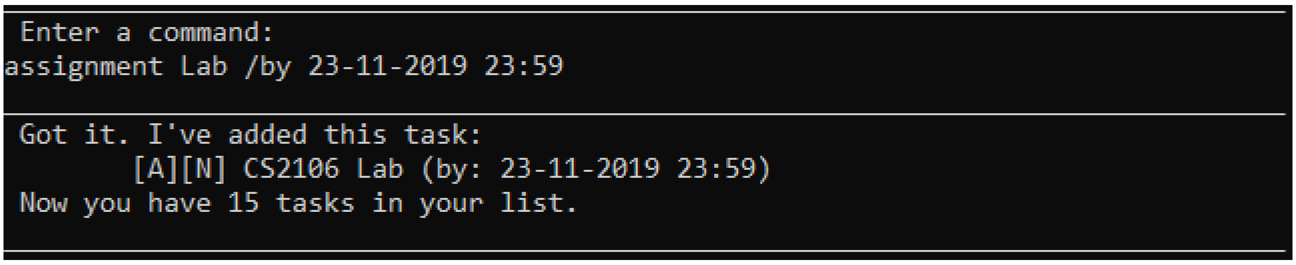
Format: `assignment DESCRIPTION /by DD-MM-YYYY HH:MM`

> `DESCRIPTION` has a character limit of 20 characters.

- `Description` of the assessment can have multiple words, not just limited to single-word descriptions.
- `Date and time` have to **strictly** be in the format as stated above.

Example:

- `assignment Lab /by 23-11-2019 23:59`



```
 Enter a command:
assignment Lab /by 23-11-2019 23:59

 Got it. I've added this task:
        [A][N] CS2106 Lab (by: 23-11-2019 23:59)
 Now you have 15 tasks in your list.
```

Figure 17. Adding an assignment

Adds an assignment `Lab` for the selected module with the due date as `23-11-2019 23:59`.

3.5.3. Adding a deadline: `deadline`

You can choose to add a task with a deadline.

Format: `deadline DESCRIPTION /by DD-MM-YYYY HH:MM`

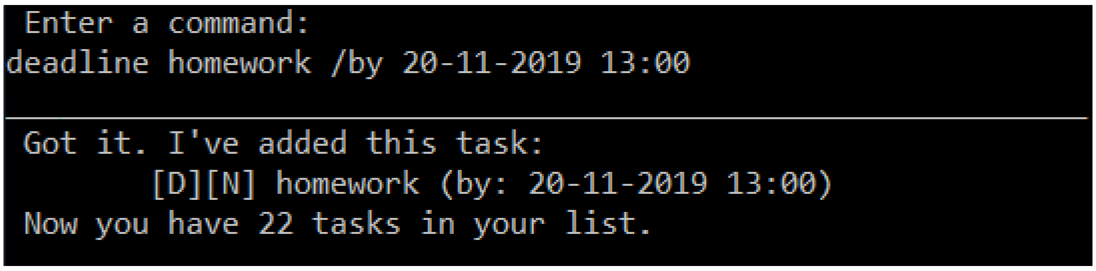
> `DESCRIPTION` has a character limit of 20 characters.

- `Description` of the task to be done can have multiple words, not just limited to single-word descriptions.

- `Date and time` have to **strictly** be in the format as stated above.

Example:

- `deadline homework /by 20-11-2019 13:00`



```
 Enter a command:
deadline homework /by 20-11-2019 13:00
____________________________________________________________
 Got it. I've added this task:
       [D][N] homework (by: 20-11-2019 13:00)
 Now you have 22 tasks in your list.
```

Figure 18. Example usage of deadline feature

Adds a task with description and datetime to be `homework` and `20-11-2019 13:00` respectively.

3.5.4. Adding an event: `event`

You can add an event with a scheduled starting and ending time.

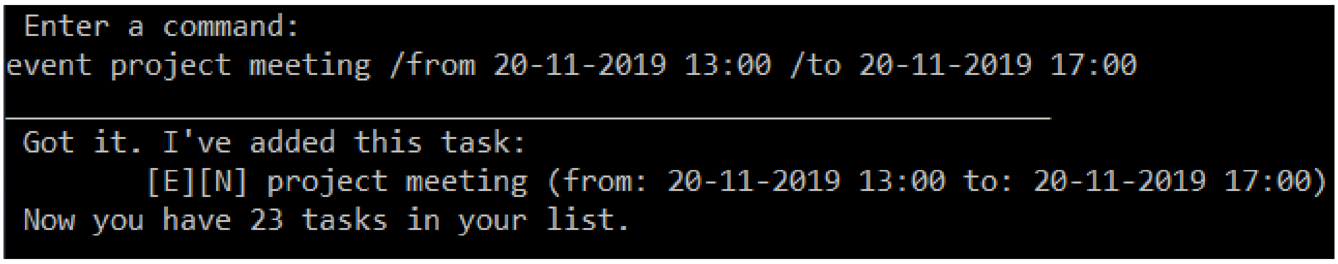
Format: `event DESCRIPTION /from DD-MM-YYYY HH:MM /to DD-MM-YYYY HH:MM`

> `DESCRIPTION` has a character limit of 20 characters.

- `Description` of the task to be done can have multiple words, not just limited to single-word descriptions.
- `Date and time` have to **strictly** be in the format as stated above.

Example:

- `event project meeting /from 20-11-2019 13:00 /to 20-11-2019 17:00`



```
 Enter a command:
event project meeting /from 20-11-2019 13:00 /to 20-11-2019 17:00
____________________________________________________________
 Got it. I've added this task:
       [E][N] project meeting (from: 20-11-2019 13:00 to: 20-11-2019 17:00)
 Now you have 23 tasks in your list.
```

Figure 19. Example usage of event feature

Adds an event with description, start and end time to be `project meeting`, `20-11-2019 13:00` and `20-11-2019 17:00` respectively.

3.5.5. Adding a todo: `todo`

You can choose to add a task to be done on a specific day.

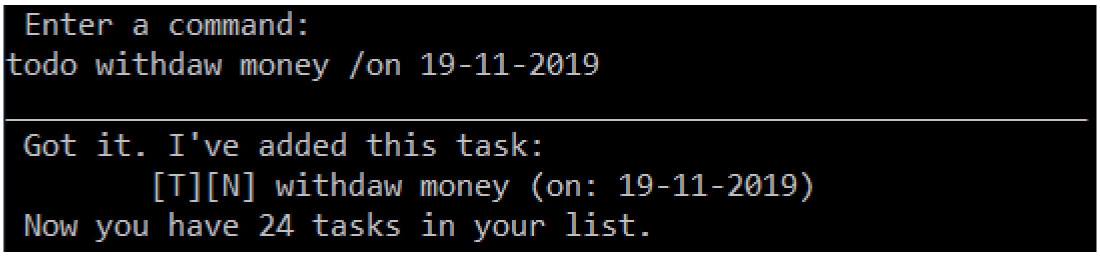
Format: `todo DESCRIPTION /on DD-MM-YYYY`

DESCRIPTION has a character limit of 20 characters.

- Description of the task to be done can have multiple words, not just limited to single-word descriptions.
- Date has to **strictly** be in the format as stated above.

Example:

- todo withdraw money /on 19-11-2019



```
 Enter a command:
todo withdaw money /on 19-11-2019
_______________________________________________________________
 Got it. I've added this task:
       [T][N] withdaw money (on: 19-11-2019)
 Now you have 24 tasks in your list.
```

Figure 20. Example usage of todo feature

Adds a task called withdraw money on 19-11-2019.

3.6. Modifying tasks

3.6.1. Setting a recurring task: recurring

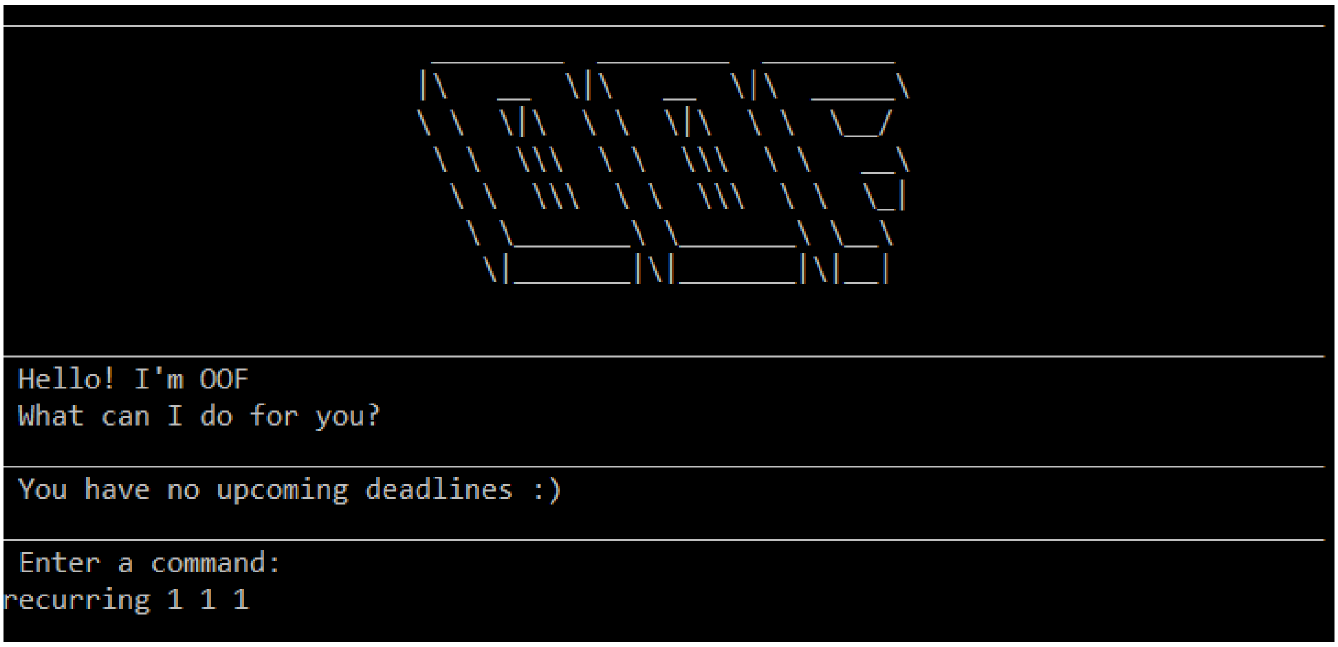
You can select a task that will be repeated based on your preference.

Format: recurring INDEX NUMBER_OF_OCCURRENCES FREQUENCY

- The INDEX refers to the index number displayed in the list of tasks recorded. (list can be used to display the saved tasks).
- NUMBER_OF_OCCURRENCES refers to the number of times the selected task recurs which is an integer from 1-10.
- FREQUENCY refers to the recurring frequency which is an integer from 1-4.
    - 1. DAILY
    - 2. WEEKLY
    - 3. MONTHLY
    - 4. YEARLY

Example:

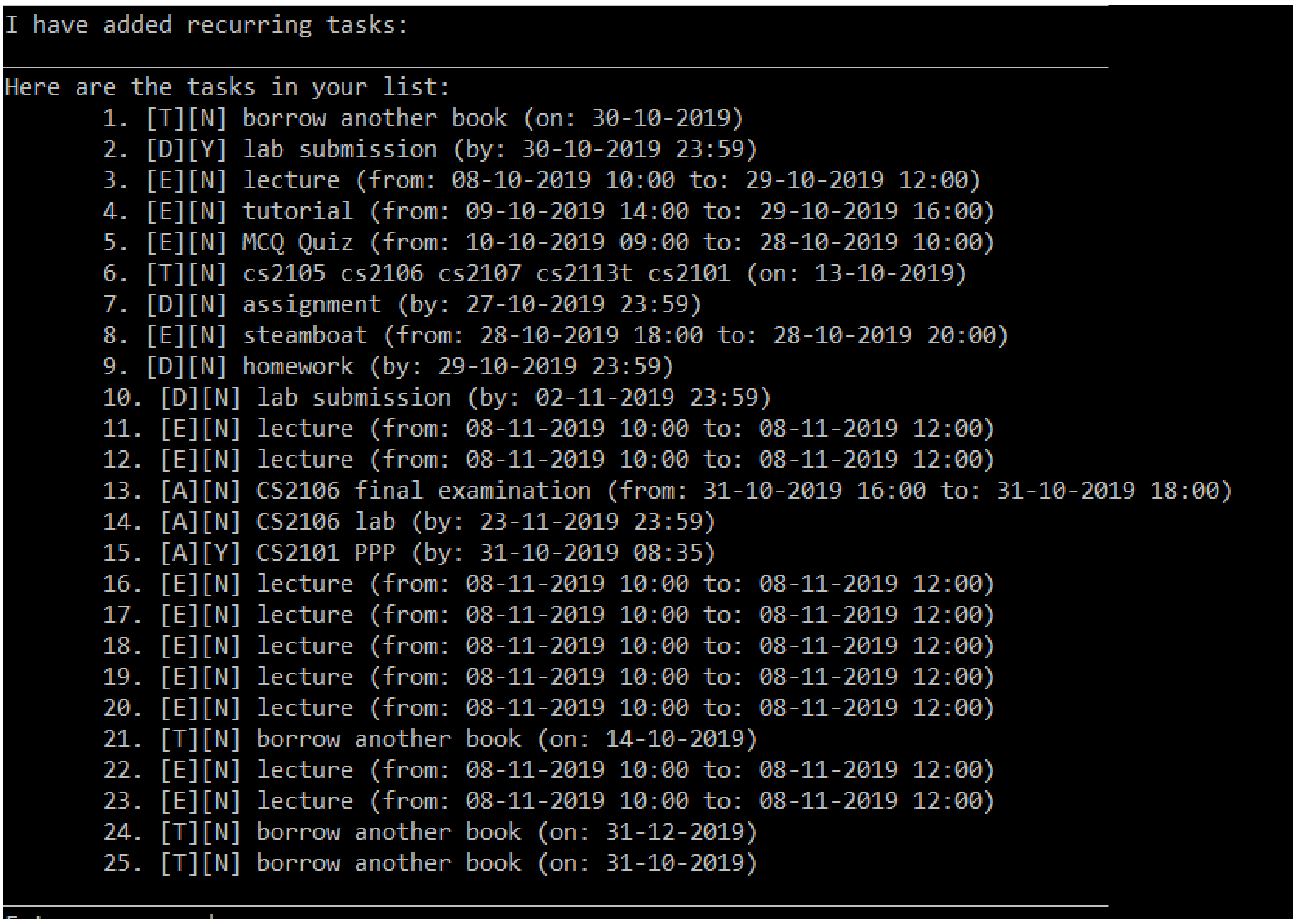
1. The user enters recurring 1 1 1



```
 ________  ________  ________
|\   __  \|\   __  \|\  _____\
\ \  \|\  \ \  \|\  \ \  \__/
 \ \  \\\  \ \  \\\  \ \   __\
  \ \  \\\  \ \  \\\  \ \  \_|
   \ \_______\ \_______\ \__\
    \|_______|\|_______|\|__|

 Hello! I'm OOF
 What can I do for you?

 You have no upcoming deadlines :)

 Enter a command:
recurring 1 1 1
```

Figure 21. Example to show recurring feature's usage

2. The user presses `ENTER`



```
I have added recurring tasks:

Here are the tasks in your list:
       1. [T][N] borrow another book (on: 30-10-2019)
       2. [D][Y] lab submission (by: 30-10-2019 23:59)
       3. [E][N] lecture (from: 08-10-2019 10:00 to: 29-10-2019 12:00)
       4. [E][N] tutorial (from: 09-10-2019 14:00 to: 29-10-2019 16:00)
       5. [E][N] MCQ Quiz (from: 10-10-2019 09:00 to: 28-10-2019 10:00)
       6. [T][N] cs2105 cs2106 cs2107 cs2113t cs2101 (on: 13-10-2019)
       7. [D][N] assignment (by: 27-10-2019 23:59)
       8. [E][N] steamboat (from: 28-10-2019 18:00 to: 28-10-2019 20:00)
       9. [D][N] homework (by: 29-10-2019 23:59)
       10. [D][N] lab submission (by: 02-11-2019 23:59)
       11. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       12. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       13. [A][N] CS2106 final examination (from: 31-10-2019 16:00 to: 31-10-2019 18:00)
       14. [A][N] CS2106 lab (by: 23-11-2019 23:59)
       15. [A][Y] CS2101 PPP (by: 31-10-2019 08:35)
       16. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       17. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       18. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       19. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       20. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       21. [T][N] borrow another book (on: 14-10-2019)
       22. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       23. [E][N] lecture (from: 08-11-2019 10:00 to: 08-11-2019 12:00)
       24. [T][N] borrow another book (on: 31-12-2019)
       25. [T][N] borrow another book (on: 31-10-2019)
```

Figure 22. Output after selecting option 2

The command shows the new recurring task that was added.

3.6.2. Marking a task as done: `done`

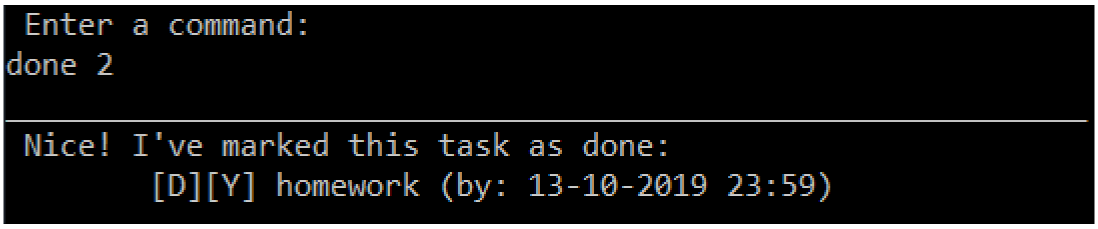
You can mark tasks as completed so that you can track your progress.

Format: `done INDEX`

- The `INDEX` refers to the index number displayed in the list of tasks recorded. (`list` can be used to display the saved tasks).

Examples:

- `done 2`



```
 Enter a command:
done 2
____________________________________________________________
 Nice! I've marked this task as done:
       [D][Y] homework (by: 13-10-2019 23:59)
```

Figure 23. Output of done command.

Marks the 2nd task in the list of tasks as done.

3.6.3. Deleting a task: `delete`

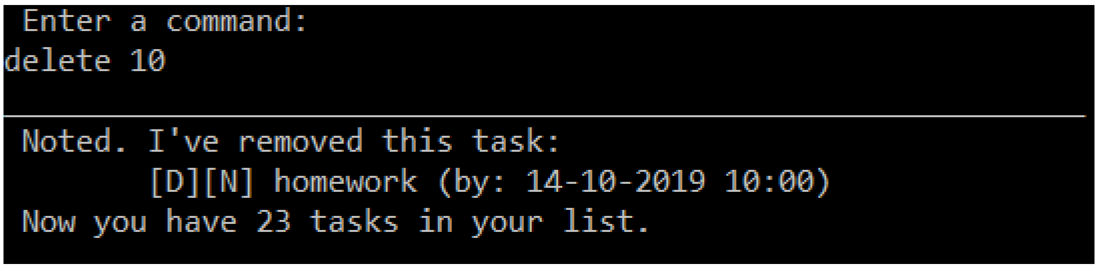
You can delete tasks that you have completed or are no longer valid.

Format: `delete INDEX`

- The `INDEX` refers to the index number displayed in the list of tasks recorded. (`list` can be used to display the saved tasks).

Examples:

- `delete 10`



```
 Enter a command:
delete 10
____________________________________________________________
 Noted. I've removed this task:
       [D][N] homework (by: 14-10-2019 10:00)
 Now you have 23 tasks in your list.
```

Figure 24. Output of delete command

Deletes the 10th task in the list of tasks.

3.7. Productivity

3.7.1. Finding tasks quickly: `find`

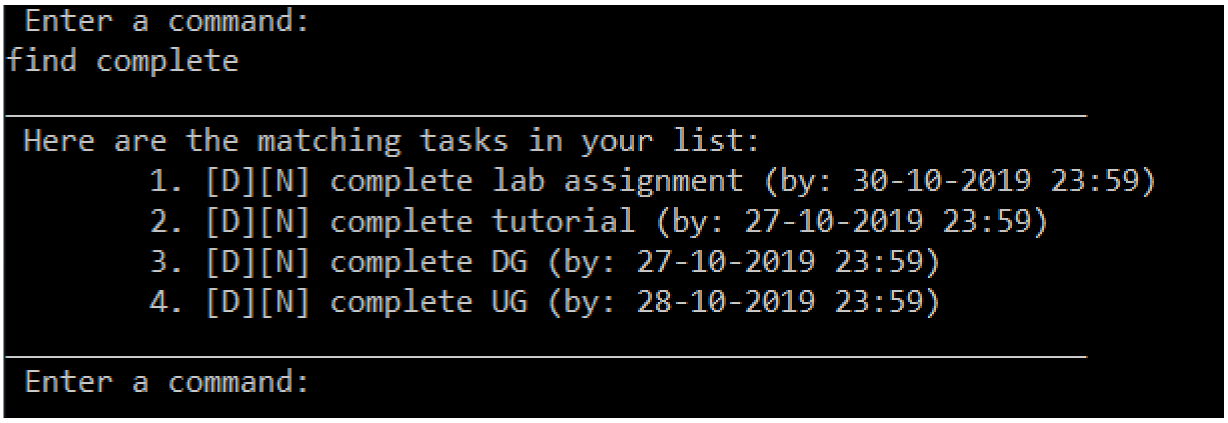
You can quickly find anything you have inputted by providing **OOF** with a keyword.

Format: `find DESCRIPTION`

- `Description` of the task to be done can have multiple words, not just limited to single-word descriptions.

Example:

- `find complete`



```
 Enter a command:
find complete
________________________________________________________________
 Here are the matching tasks in your list:
        1. [D][N] complete lab assignment (by: 30-10-2019 23:59)
        2. [D][N] complete tutorial (by: 27-10-2019 23:59)
        3. [D][N] complete DG (by: 27-10-2019 23:59)
        4. [D][N] complete UG (by: 28-10-2019 23:59)
________________________________________________________________
 Enter a command:
```

Figure 25. Output of find command

Finds tasks with `complete` in the description.

3.7.2. Viewing free time slots: `free`

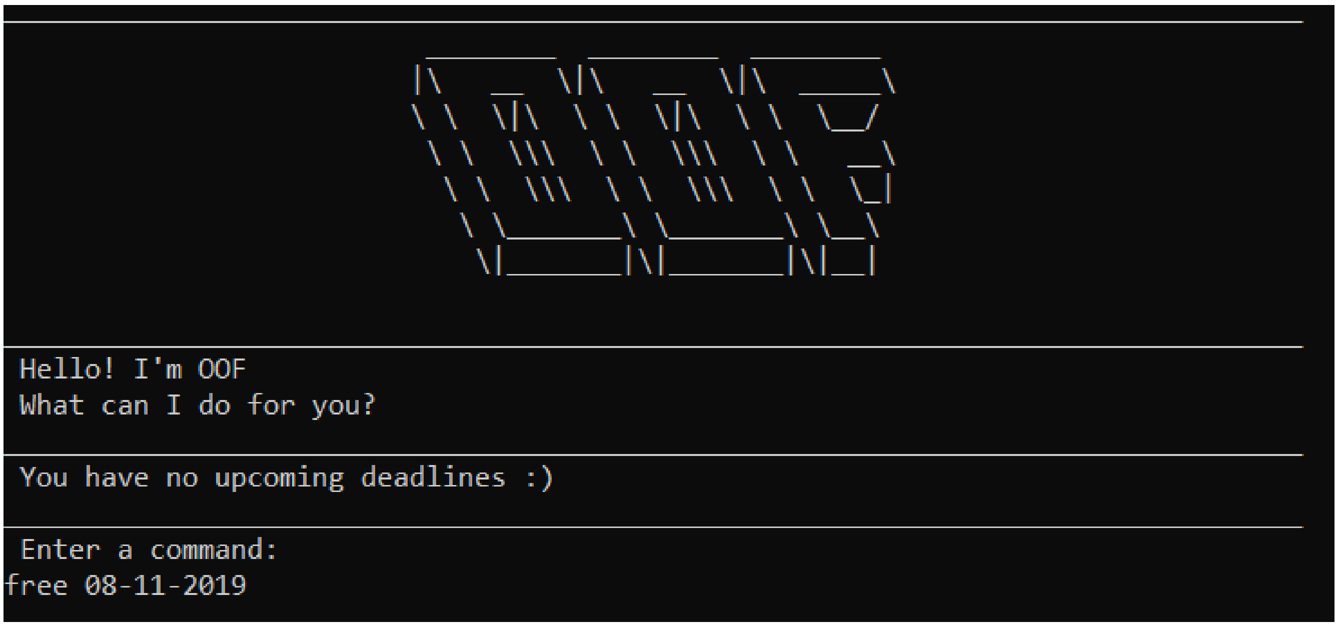
You can view the time slots you are available on a specific day so that you can plan project meetings with your friends. You can also receive suggestions on which deadlines to complete in your free time if they are due one week from the date specified.

Format: `free DATE`

- `DATE` has to **strictly** be in the format `DD-MM-YYYY`.

Example:

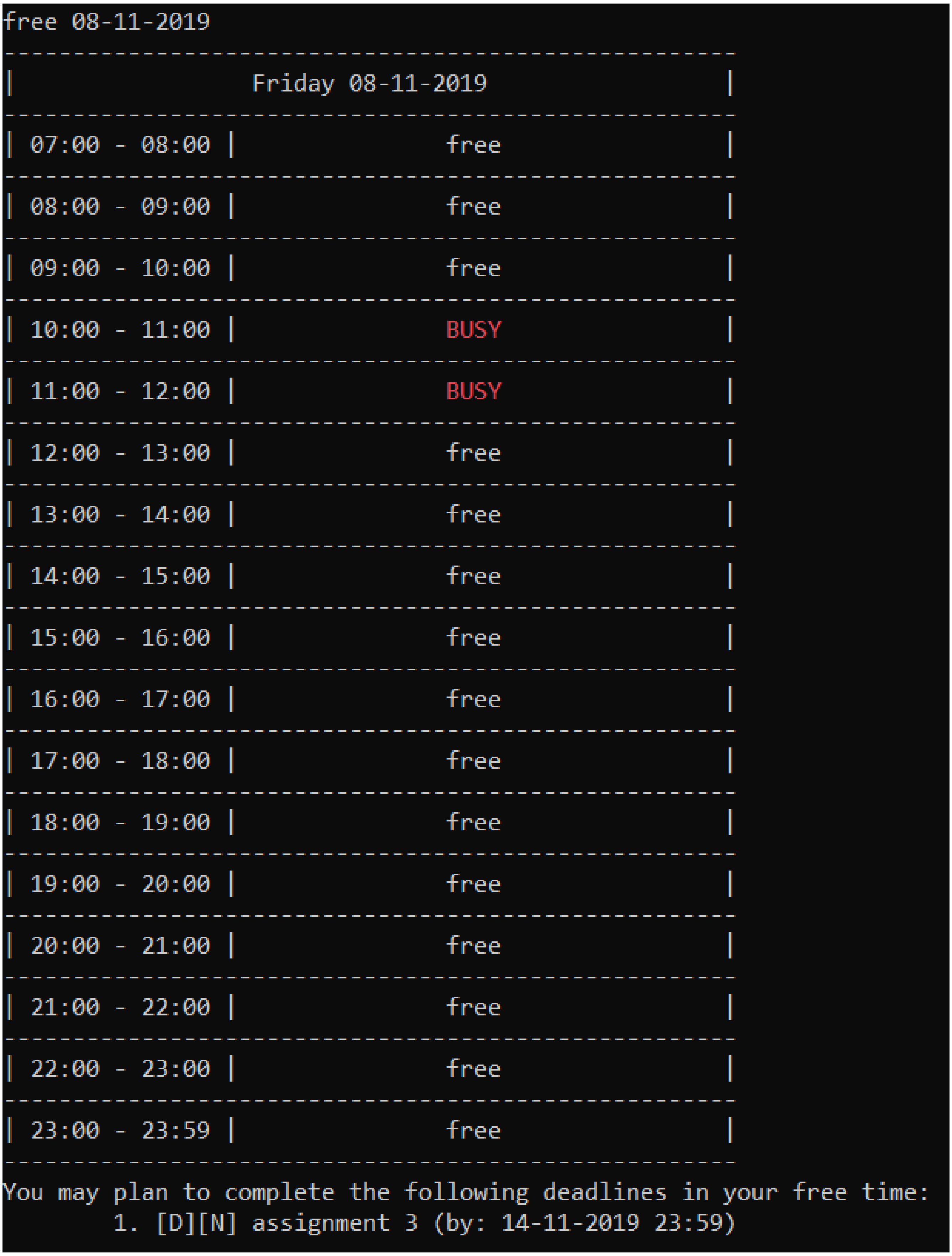
1. Type `free 08-11-2019` as a command press `ENTER`



```
 ________  ________  ________
|\   __  \|\   __  \|\  _____\
\ \  \|\  \ \  \|\  \ \  \__/
 \ \  \\\  \ \  \\\  \ \   __\
  \ \  \\\  \ \  \\\  \ \  \_|
   \ \_______\ \_______\ \__\
    \|_______|\|_______|\|__|

 Hello! I'm OOF
 What can I do for you?

 You have no upcoming deadlines :)

 Enter a command:
free 08-11-2019
```

Figure 26. Typing free 08-11-2019 into OOF

2. `OOF` displays all the free time slots that you have followed by a suggestion if there are upcoming deadlines due.

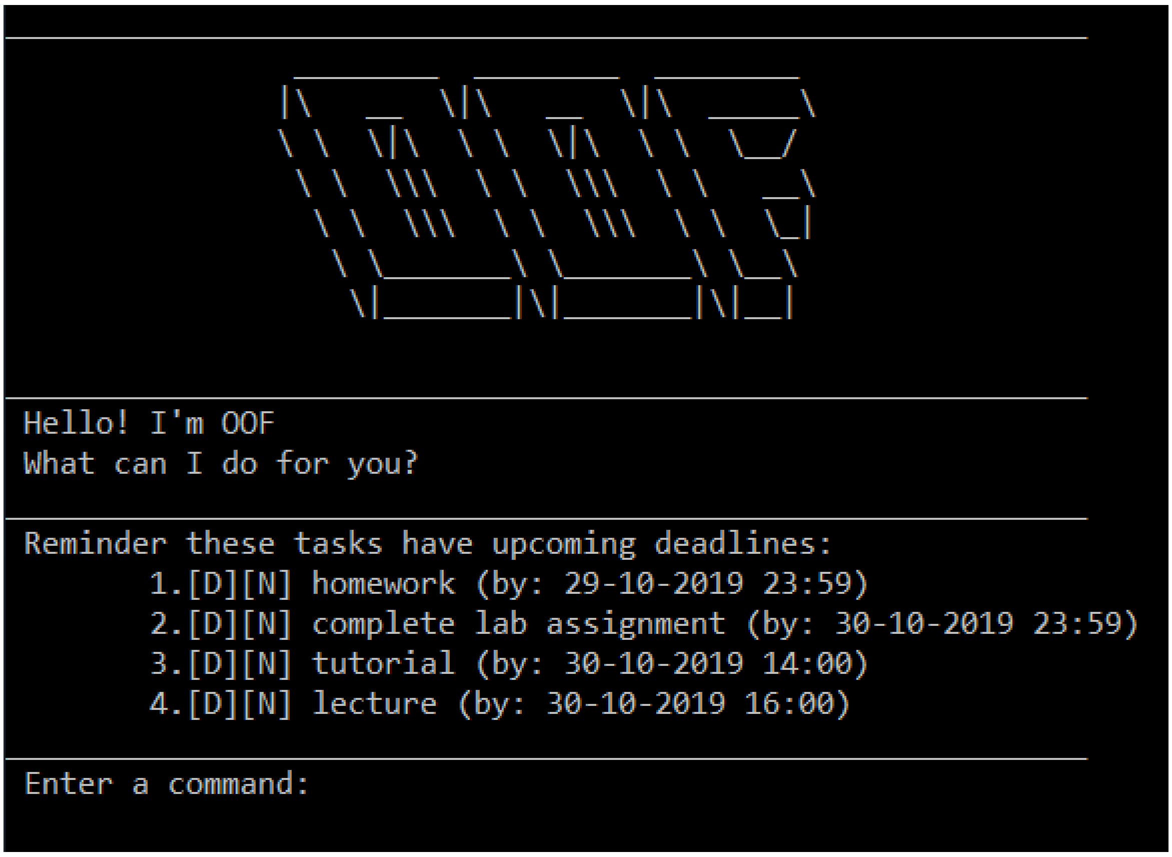


```
free 08-11-2019
-----------------------------------------------------
|                 Friday 08-11-2019                 |
-----------------------------------------------------
| 07:00 - 08:00 |               free                |
-----------------------------------------------------
| 08:00 - 09:00 |               free                |
-----------------------------------------------------
| 09:00 - 10:00 |               free                |
-----------------------------------------------------
| 10:00 - 11:00 |               BUSY                |
-----------------------------------------------------
| 11:00 - 12:00 |               BUSY                |
-----------------------------------------------------
| 12:00 - 13:00 |               free                |
-----------------------------------------------------
| 13:00 - 14:00 |               free                |
-----------------------------------------------------
| 14:00 - 15:00 |               free                |
-----------------------------------------------------
| 15:00 - 16:00 |               free                |
-----------------------------------------------------
| 16:00 - 17:00 |               free                |
-----------------------------------------------------
| 17:00 - 18:00 |               free                |
-----------------------------------------------------
| 18:00 - 19:00 |               free                |
-----------------------------------------------------
| 19:00 - 20:00 |               free                |
-----------------------------------------------------
| 20:00 - 21:00 |               free                |
-----------------------------------------------------
| 21:00 - 22:00 |               free                |
-----------------------------------------------------
| 22:00 - 23:00 |               free                |
-----------------------------------------------------
| 23:00 - 23:59 |               free                |
-----------------------------------------------------
You may plan to complete the following deadlines in your free time:
        1. [D][N] assignment 3 (by: 14-11-2019 23:59)
```

Figure 27. Typing free with a valid date in the valid format of DD-MM-YYYY

3.7.3. Setting reminders for upcoming deadlines: `NIL`

You can get timely reminders for the tasks that are expiring.



```
 ________  ________  ________
|\   __  \|\   __  \|\  _____\
\ \  \|\  \ \  \|\  \ \  \__/
 \ \  \\\  \ \  \\\  \ \   __\
  \ \  \\\  \ \  \\\  \ \  \_|
   \ \_______\ \_______\ \__\
    \|_______|\|_______|\|__|

Hello! I'm OOF
What can I do for you?

Reminder these tasks have upcoming deadlines:
        1.[D][N] homework (by: 29-10-2019 23:59)
        2.[D][N] complete lab assignment (by: 30-10-2019 23:59)
        3.[D][N] tutorial (by: 30-10-2019 14:00)
        4.[D][N] lecture (by: 30-10-2019 16:00)

Enter a command:
```

Figure 28. Output of reminder command

This command functions in the background so **OOF** automatically reminds you of the expiring tasks when you start our program.
You can customise the `threshold` to tell **OOF** when to remind you to complete your tasks.
View the detailed description on the usage of `threshold` below.

3.7.4. Choosing a threshold for tasks: `threshold`

You can set a comfortable threshold to tell **OOF** when to remind you to complete your tasks.

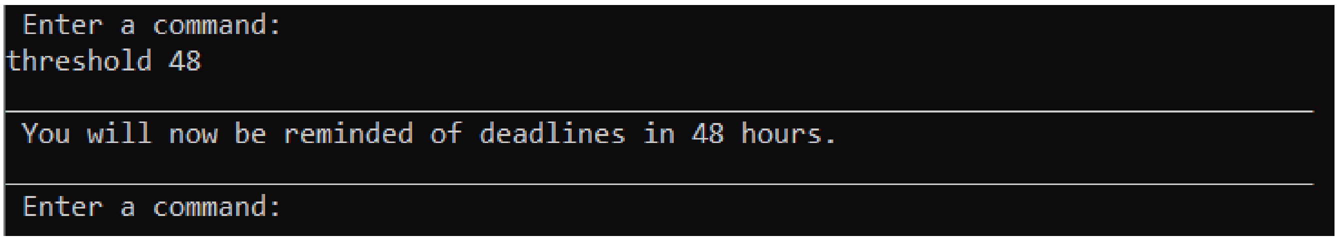
Format: `threshold HH`

- `HH` represents the minimum number of hours from the `current time` to the `deadline` of tasks before **OOF** reminds you of those tasks.

Example:

- `threshold 48`

Example:



```
 Enter a command:
threshold 48
________________________________________________________________________________
 You will now be reminded of deadlines in 48 hours.
________________________________________________________________________________
 Enter a command:
```

Figure 29. Output of threshold command

Tasks that have `deadlines` within 48 hours from the `current time` will be in the reminders.

3.7.5. Starting/Stopping/Pausing Task Tracker: `tracker`

You can track a task from the current time.

Format: `tracker /INSTRUCTION TASK_INDEX MODULE_CODE`

Options for INSTRUCTION

start

begin tracking a task from the current time.

pause

pause tracking a task from the current time.

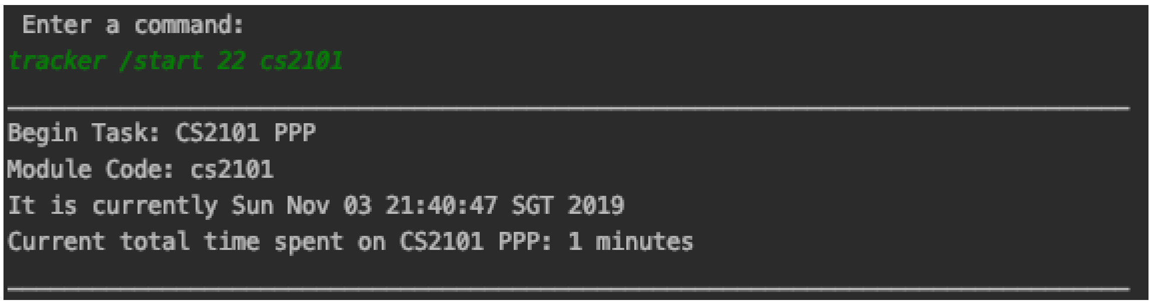
stop

stop tracking a task from the current time.

view

view a histogram featuring the total amount fo time spent on each module.

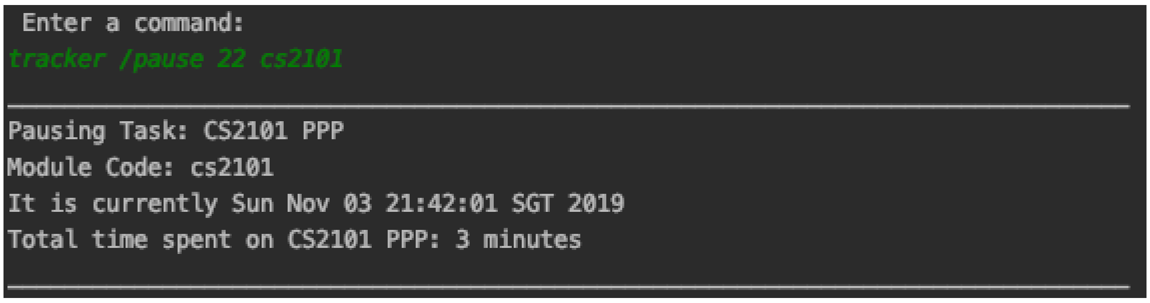
Example: `tracker /start 22 cs2101`



```
 Enter a command:
tracker /start 22 cs2101
________________________________________________________________________________
Begin Task: CS2101 PPP
Module Code: cs2101
It is currently Sun Nov 03 21:40:47 SGT 2019
Current total time spent on CS2101 PPP: 1 minutes
________________________________________________________________________________
```

Figure 30. Starts Task Tracker

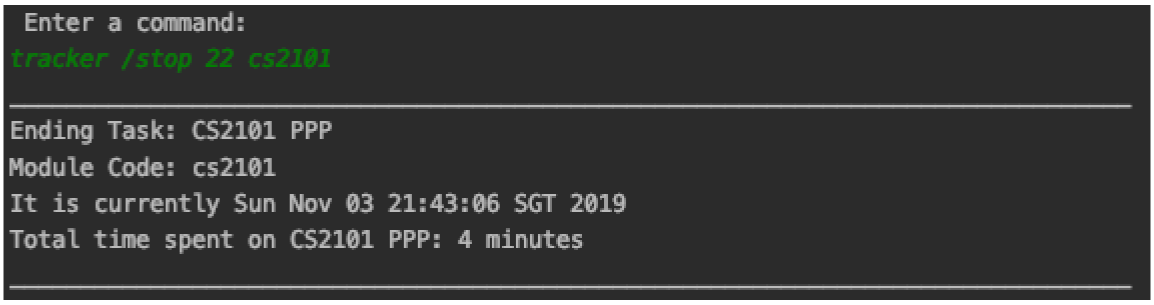
Example: `tracker /pause 22 cs2101`



```
 Enter a command:
tracker /pause 22 cs2101
________________________________________________________________________________
Pausing Task: CS2101 PPP
Module Code: cs2101
It is currently Sun Nov 03 21:42:01 SGT 2019
Total time spent on CS2101 PPP: 3 minutes
________________________________________________________________________________
```

Figure 31. Pauses Task Tracker

Example: `tracker /stop 22 cs2101`



```
 Enter a command:
tracker /stop 22 cs2101
________________________________________________________________________________
Ending Task: CS2101 PPP
Module Code: cs2101
It is currently Sun Nov 03 21:43:06 SGT 2019
Total time spent on CS2101 PPP: 4 minutes
________________________________________________________________________________
```

Figure 32. Stops Task Tracker

3.7.6. Viewing Task Tracker: `tracker`

You can view a histogram featuring the amount of time you spend on each module in blocks of 10 minutes.

Format: `tracker /view PERIOD`

Options for TIME_PERIOD

day

filter time spent on each `Module` today.

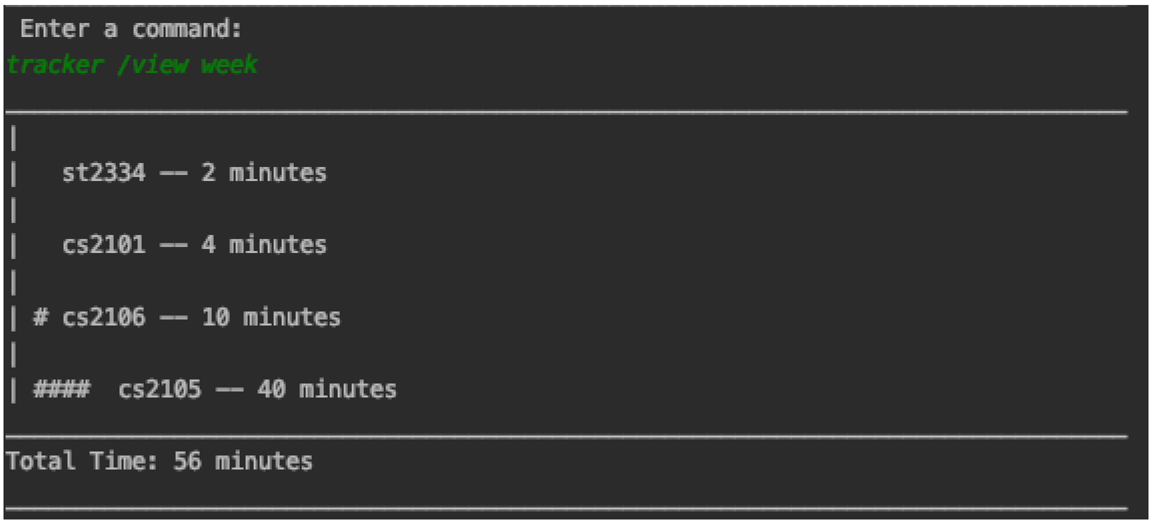
week

filter time spent on each `Module` over the last 7 days.

all

filter time spent on each `Module` over all entries.

Example: `tracker /view week`



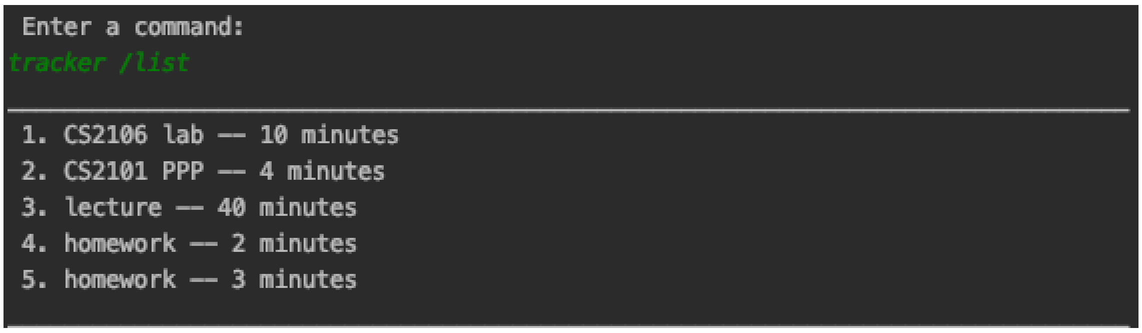
```
Enter a command:
tracker /view week
________________________________________________________________
|
|   st2334 -- 2 minutes
|
|   cs2101 -- 4 minutes
|
| # cs2106 -- 10 minutes
|
| ####  cs2105 -- 40 minutes
________________________________________________________________
Total Time: 56 minutes
________________________________________________________________
```

Figure 33. Displays Task Tracker by Module Code

3.7.7. View Task Tracker List: `tracker`

You can view a list of all your Task trackers.

Format: `tracker /list`



```
Enter a command:
tracker /list
________________________________________________________________
1. CS2106 lab -- 10 minutes
2. CS2101 PPP -- 4 minutes
3. lecture -- 40 minutes
4. homework -- 2 minutes
5. homework -- 3 minutes
________________________________________________________________
```

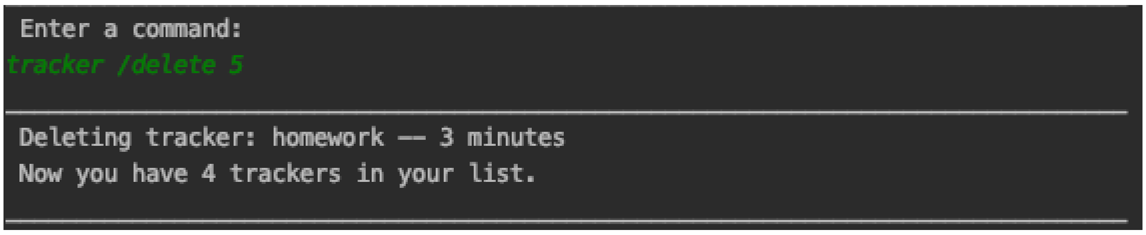
Figure 34. Displays a list of Task Trackers

3.7.8. Delete a Task Tracker: `tracker`

You can delete a Task Tracker.

Format: `tracker /delete TRACKER_INDEX`

Example: `tracker /delete `



```
Enter a command:
tracker /delete 5
________________________________________________________________
Deleting tracker: homework -- 3 minutes
Now you have 4 trackers in your list.
________________________________________________________________
```

Figure 35. Deletes a Task Tracker

3.8. Organisation

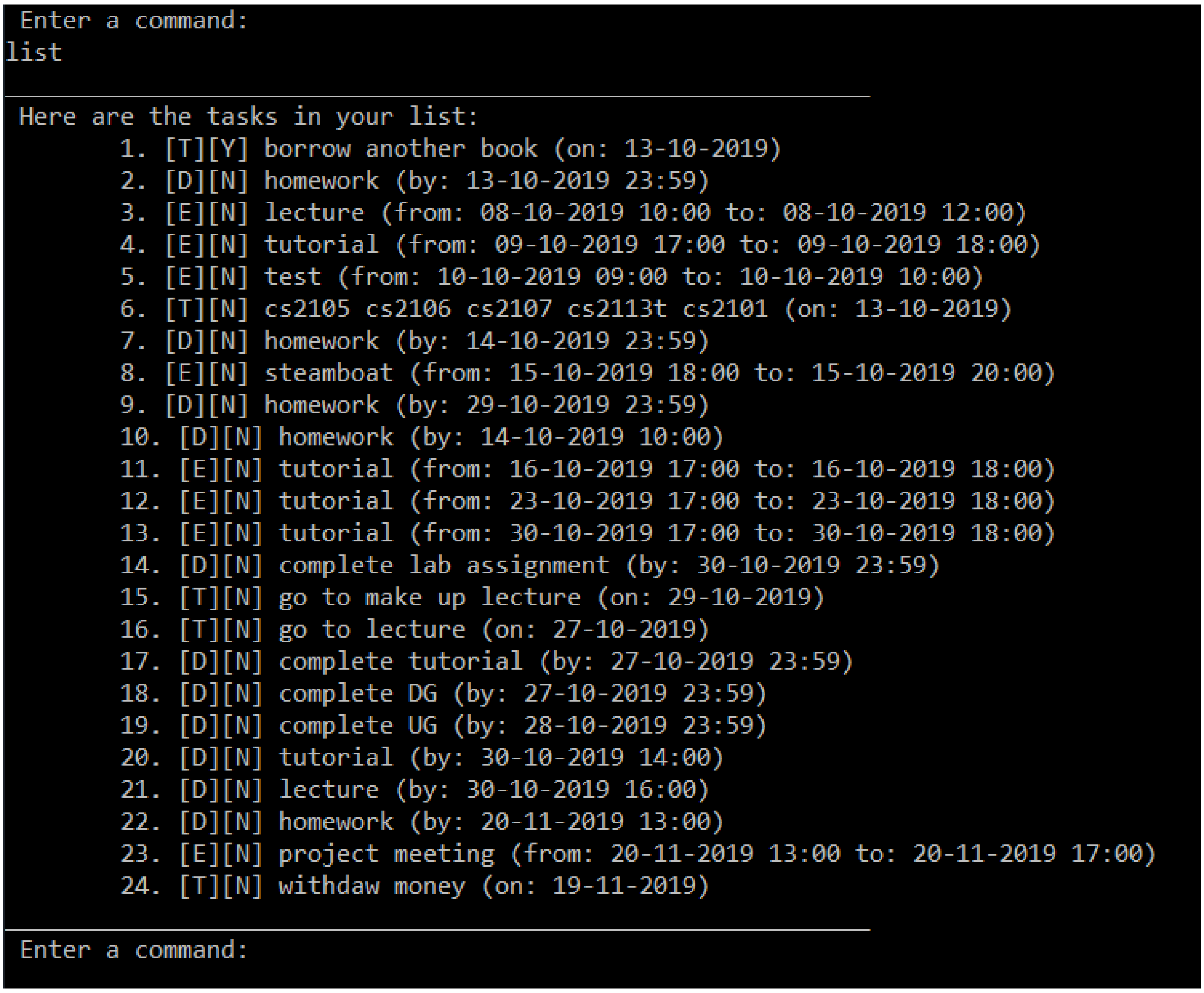
3.8.1. Listing tasks: `list`

You can list all the tasks that you have saved in **OOF**.

Format: `list`

Example:

- User enters `list`



```
 Enter a command:
list
____________________________________________________________
 Here are the tasks in your list:
        1. [T][Y] borrow another book (on: 13-10-2019)
        2. [D][N] homework (by: 13-10-2019 23:59)
        3. [E][N] lecture (from: 08-10-2019 10:00 to: 08-10-2019 12:00)
        4. [E][N] tutorial (from: 09-10-2019 17:00 to: 09-10-2019 18:00)
        5. [E][N] test (from: 10-10-2019 09:00 to: 10-10-2019 10:00)
        6. [T][N] cs2105 cs2106 cs2107 cs2113t cs2101 (on: 13-10-2019)
        7. [D][N] homework (by: 14-10-2019 23:59)
        8. [E][N] steamboat (from: 15-10-2019 18:00 to: 15-10-2019 20:00)
        9. [D][N] homework (by: 29-10-2019 23:59)
        10. [D][N] homework (by: 14-10-2019 10:00)
        11. [E][N] tutorial (from: 16-10-2019 17:00 to: 16-10-2019 18:00)
        12. [E][N] tutorial (from: 23-10-2019 17:00 to: 23-10-2019 18:00)
        13. [E][N] tutorial (from: 30-10-2019 17:00 to: 30-10-2019 18:00)
        14. [D][N] complete lab assignment (by: 30-10-2019 23:59)
        15. [T][N] go to make up lecture (on: 29-10-2019)
        16. [T][N] go to lecture (on: 27-10-2019)
        17. [D][N] complete tutorial (by: 27-10-2019 23:59)
        18. [D][N] complete DG (by: 27-10-2019 23:59)
        19. [D][N] complete UG (by: 28-10-2019 23:59)
        20. [D][N] tutorial (by: 30-10-2019 14:00)
        21. [D][N] lecture (by: 30-10-2019 16:00)
        22. [D][N] homework (by: 20-11-2019 13:00)
        23. [E][N] project meeting (from: 20-11-2019 13:00 to: 20-11-2019 17:00)
        24. [T][N] withdaw money (on: 19-11-2019)
____________________________________________________________
 Enter a command:
```

Figure 36. Output of list command

A list of tasks currently saved in **OOF** will be displayed.

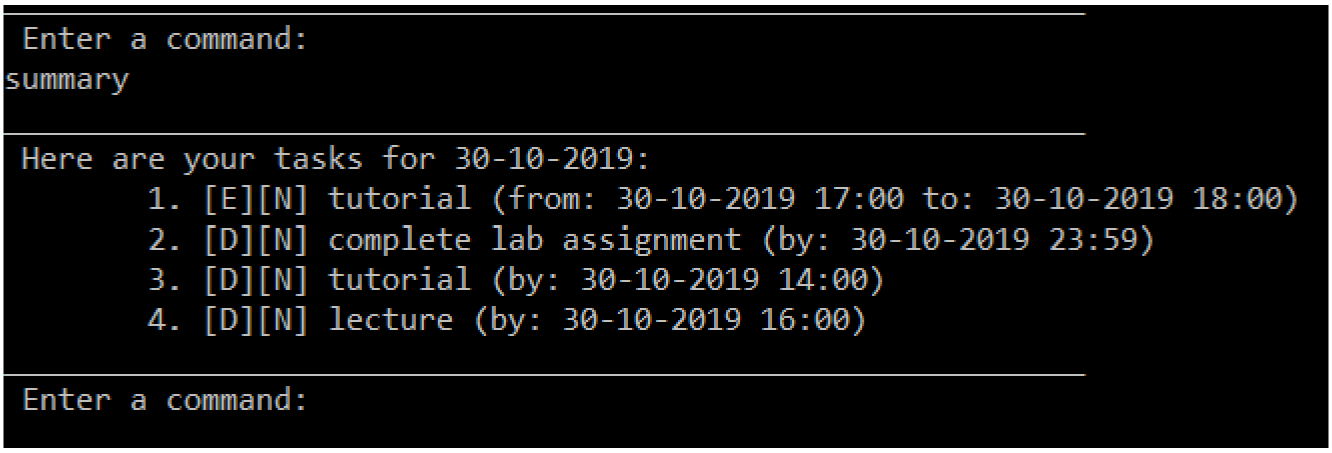
3.8.2. Viewing a summary of the next day’s task: `summary`

You can view a summary of all the tasks to be done for the next day.

Format: `summary`

Example:

- `summary`



```
 Enter a command:
summary

 Here are your tasks for 30-10-2019:
        1. [E][N] tutorial (from: 30-10-2019 17:00 to: 30-10-2019 18:00)
        2. [D][N] complete lab assignment (by: 30-10-2019 23:59)
        3. [D][N] tutorial (by: 30-10-2019 14:00)
        4. [D][N] lecture (by: 30-10-2019 16:00)

 Enter a command:
```

Figure 37. Output of summary command

Provides a summary of a list of todo, deadlines and events that will occur tomorrow.

3.8.3. Viewing a summary of a day's task by date: `schedule`

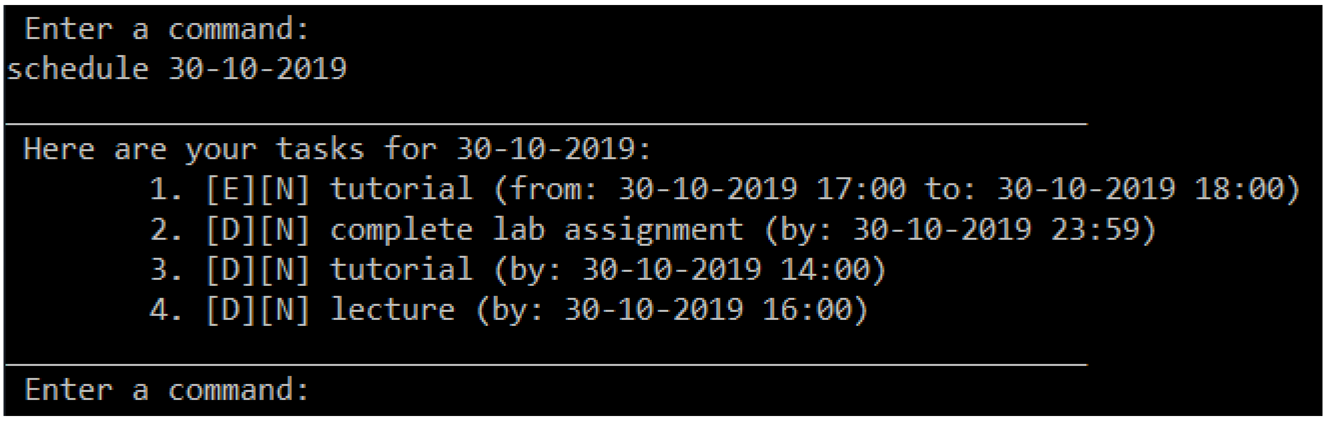
You can view a summary of all the tasks and events on a specific day of your choice.

Format: `schedule DD-MM-YYYY`

- `Date` has to strictly be in the format as stated above.

Example:

- `schedule 30-10-2019`



```
 Enter a command:
schedule 30-10-2019

 Here are your tasks for 30-10-2019:
        1. [E][N] tutorial (from: 30-10-2019 17:00 to: 30-10-2019 18:00)
        2. [D][N] complete lab assignment (by: 30-10-2019 23:59)
        3. [D][N] tutorial (by: 30-10-2019 14:00)
        4. [D][N] lecture (by: 30-10-2019 16:00)

 Enter a command:
```

Figure 38. Output of schedule command

Provides a summary of a list of todo, deadlines and events that will occur on `30-10-2019`.

3.8.4. Viewing tasks in week view: `viewweek`

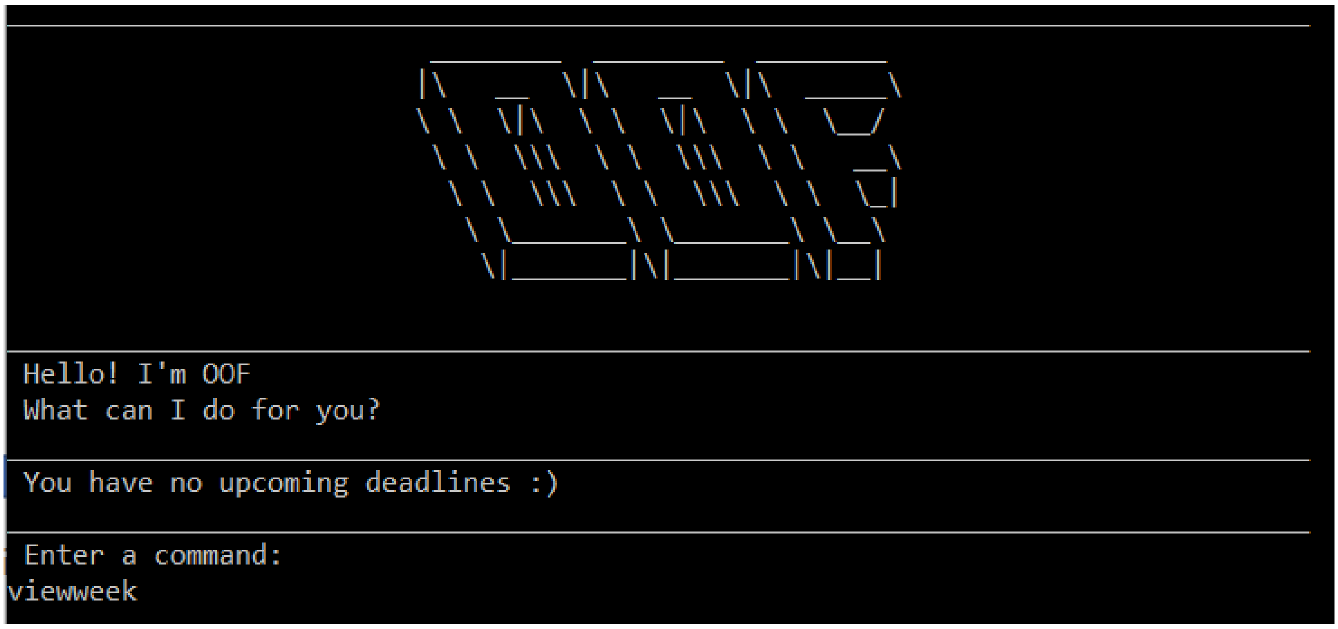
You can view the tasks for any particular week in a table format so that you can have a grasp of what to expect for a particular or even track your own progress.

Format: `viewweek DD MM YYYY`

Note that the parameters DD MM YYYY are optional and the command will automatically show tasks for the current week if these parameters are not shown. The tasks for each day are chronologically sorted.

Example:

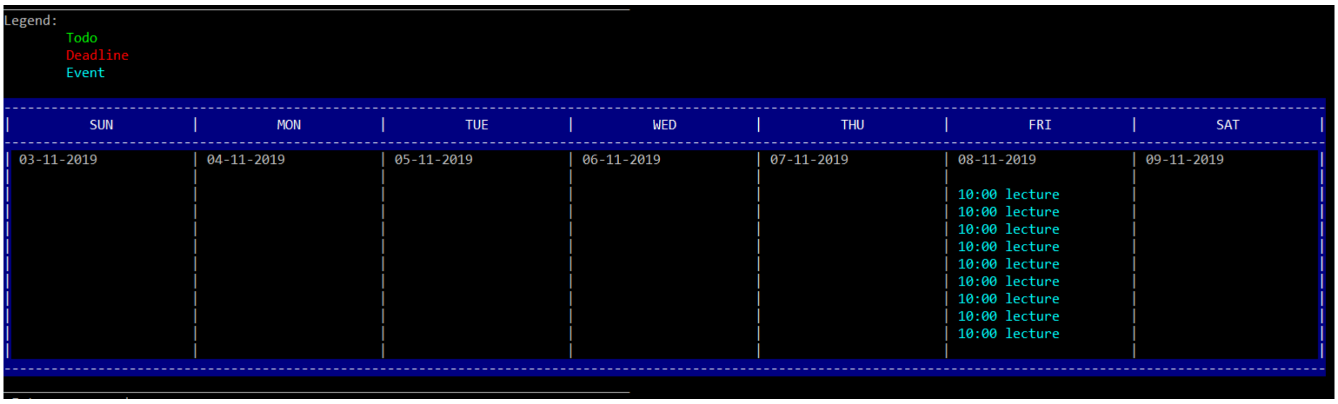
1. Type viewweek as a command and press ENTER



```
 ________  ________  ________
|\   __  \|\   __  \|\  _____\
\ \  \|\  \ \  \|\  \ \  \__/
 \ \  \\\  \ \  \\\  \ \   __\
  \ \  \\\  \ \  \\\  \ \  \_|
   \ \_______\ \_______\ \__\
    \|_______|\|_______|\|__|

 Hello! I'm OOF
 What can I do for you?

 You have no upcoming deadlines :)

 Enter a command:
viewweek
```

Figure 39. Typing viewweek into OOF

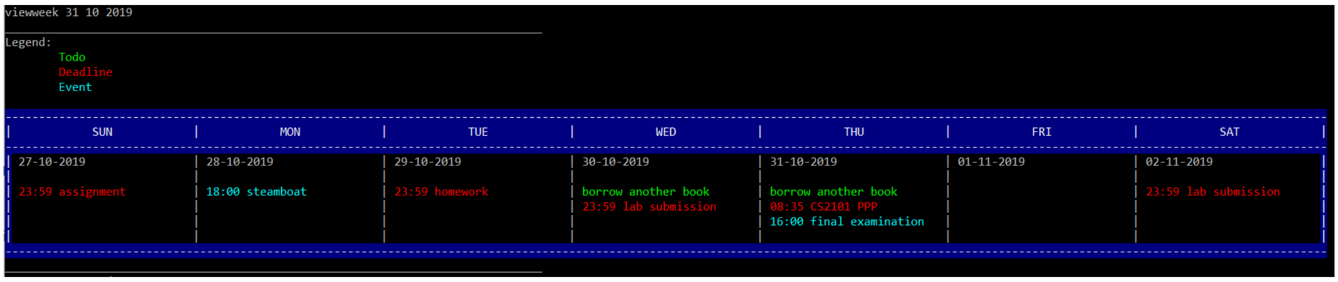
2. OOF displays tasks for the week for you.



```
Legend:
        Todo
        Deadline
        Event

-----------------------------------------------------------------------------------------------------------------------------
|          SUN          |          MON          |          TUE          |          WED          |          THU          |          FRI          |          SAT          |
-----------------------------------------------------------------------------------------------------------------------------
| 03-11-2019            | 04-11-2019            | 05-11-2019            | 06-11-2019            | 07-11-2019            | 08-11-2019            | 09-11-2019            |
|                       |                       |                       |                       |                       |                       |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       | 10:00 lecture         |                       |
|                       |                       |                       |                       |                       |                       |                       |
-----------------------------------------------------------------------------------------------------------------------------
```

Figure 40. Typing viewweek without date

3. If you wish to display tasks for a particular week, you can input DD MM YYYY.



```
viewweek 31 10 2019

Legend:
        Todo
        Deadline
        Event

-----------------------------------------------------------------------------------------------------------------------------
|          SUN          |          MON          |          TUE          |          WED          |          THU          |          FRI          |          SAT          |
-----------------------------------------------------------------------------------------------------------------------------
| 27-10-2019            | 28-10-2019            | 29-10-2019            | 30-10-2019            | 31-10-2019            | 01-11-2019            | 02-11-2019            |
|                       |                       |                       |                       |                       |                       |                       |
| 23:59 assignment      | 18:00 steamboat       | 23:59 homework        | borrow another book   | borrow another book   |                       | 23:59 lab submission  |
|                       |                       |                       | 23:59 lab submission  | 08:35 CS2101 PPP      |                       |                       |
|                       |                       |                       |                       | 16:00 final examination |                     |                       |
|                       |                       |                       |                       |                       |                       |                       |
-----------------------------------------------------------------------------------------------------------------------------
```

Figure 41. Typing viewweek with date

3.8.5. Viewing all tasks in calendar view: `calendar`

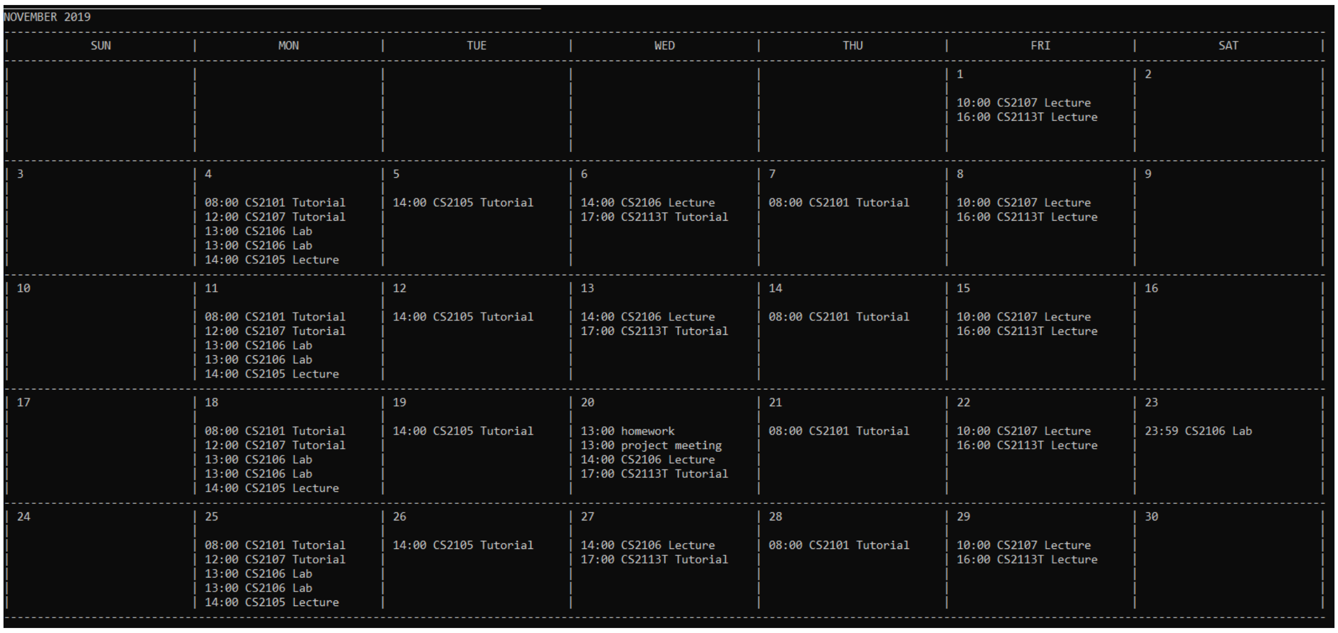
You can view all your tasks for any month so that you are aware of your schedule for that month.

Format: `calendar MM YYYY`

- `MONTH` is an integer from 1-12 (representing January to December).
- `YEAR` is an integer greater than or equal to 0.

Note that if `MONTH` and `YEAR` arguments are invalid (e.g. `calendar 13 2019`) or missing (e.g. `calendar`), the calendar for the current month and year (according to system settings) will be displayed

Example: `calendar 11 2019`



```
NOVEMBER 2019
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|          SUN           |          MON           |          TUE           |          WED           |          THU           |          FRI           |          SAT           |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|                        |                        |                        |                        |                        | 1                      | 2                      |
|                        |                        |                        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        | 10:00 CS2107 Lecture   |                        |
|                        |                        |                        |                        |                        | 16:00 CS2113T Lecture  |                        |
|                        |                        |                        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        |                        |                        |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| 3                      | 4                      | 5                      | 6                      | 7                      | 8                      | 9                      |
|                        |                        |                        |                        |                        |                        |                        |
|                        | 08:00 CS2101 Tutorial  | 14:00 CS2105 Tutorial  | 14:00 CS2106 Lecture   | 08:00 CS2101 Tutorial  | 10:00 CS2107 Lecture   |                        |
|                        | 12:00 CS2107 Tutorial  |                        | 17:00 CS2113T Tutorial |                        | 16:00 CS2113T Lecture  |                        |
|                        | 13:00 CS2106 Lab       |                        |                        |                        |                        |                        |
|                        | 13:00 CS2106 Lab       |                        |                        |                        |                        |                        |
|                        | 14:00 CS2105 Lecture   |                        |                        |                        |                        |                        |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| 10                     | 11                     | 12                     | 13                     | 14                     | 15                     | 16                     |
|                        |                        |                        |                        |                        |                        |                        |
|                        | 08:00 CS2101 Tutorial  | 14:00 CS2105 Tutorial  | 14:00 CS2106 Lecture   | 08:00 CS2101 Tutorial  | 10:00 CS2107 Lecture   |                        |
|                        | 12:00 CS2107 Tutorial  |                        | 17:00 CS2113T Tutorial |                        | 16:00 CS2113T Lecture  |                        |
|                        | 13:00 CS2106 Lab       |                        |                        |                        |                        |                        |
|                        | 13:00 CS2106 Lab       |                        |                        |                        |                        |                        |
|                        | 14:00 CS2105 Lecture   |                        |                        |                        |                        |                        |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| 17                     | 18                     | 19                     | 20                     | 21                     | 22                     | 23                     |
|                        |                        |                        |                        |                        |                        |                        |
|                        | 08:00 CS2101 Tutorial  | 14:00 CS2105 Tutorial  | 13:00 homework         | 08:00 CS2101 Tutorial  | 10:00 CS2107 Lecture   | 23:59 CS2106 Lab       |
|                        | 12:00 CS2107 Tutorial  |                        | 13:00 project meeting  |                        | 16:00 CS2113T Lecture  |                        |
|                        | 13:00 CS2106 Lab       |                        | 14:00 CS2106 Lecture   |                        |                        |                        |
|                        | 13:00 CS2106 Lab       |                        | 17:00 CS2113T Tutorial |                        |                        |                        |
|                        | 14:00 CS2105 Lecture   |                        |                        |                        |                        |                        |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| 24                     | 25                     | 26                     | 27                     | 28                     | 29                     | 30                     |
|                        |                        |                        |                        |                        |                        |                        |
|                        | 08:00 CS2101 Tutorial  | 14:00 CS2105 Tutorial  | 14:00 CS2106 Lecture   | 08:00 CS2101 Tutorial  | 10:00 CS2107 Lecture   |                        |
|                        | 12:00 CS2107 Tutorial  |                        | 17:00 CS2113T Tutorial |                        | 16:00 CS2113T Lecture  |                        |
|                        | 13:00 CS2106 Lab       |                        |                        |                        |                        |                        |
|                        | 13:00 CS2106 Lab       |                        |                        |                        |                        |                        |
|                        | 14:00 CS2105 Lecture   |                        |                        |                        |                        |                        |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

Figure 42. Viewing Calendar for month of November 2019

3.9. *Coming soon in v2.0*

3.9.1. Viewing incomplete tasks: `undone`

You can view the list of all the tasks not done that were brought forward to the next day.

Format: `undone`

Example:

- `undone` You can postpone the tasks that were not fulfilled to the next day.

3.9.2. Filtering tasks by categories: `filter`

You can filter tasks by matching the category given.

Format: `filter CATEGORY`

- `Category` of the task can be any one of the following: todo, deadline, event, recurring.

Example:

- `filter todo`
  You can display all todo tasks.

3.9.3. Adding a task: `tentative`

You can add a task that can be confirmed at a later time.

Format: `tentative DESCRIPTION`

- `Description` of the task to be done can have multiple words, not just limited to single-word descriptions.

Example:

- `tentative group lunch`
  Adds a tentative task called `group lunch`.

3.9.4. Adding a task: `do-after`

You can add a task that needs to be done after a specified task.

Format: `do-after INDEX DESCRIPTION`

- The `INDEX` refers to the index number displayed in the list of tasks recorded. (`list` can be used to display the saved tasks).
- `Description` of the task to be done can have multiple words, not just limited to single-word descriptions.

Example:

- `do-after 1 buy groceries`
  Adds a do-after task called `buy groceries` that will be displayed once the 1st task in the list has been completed.

3.9.5. Adding a task: `range`

You can add a task that needs to be completed within a certain time period

Format: `range DESCRIPTION /from DD-MM-YYYY HH:MM /to DD-MM-YYYY HH:MM`

- `Description` of the task to be done can have multiple words, not just limited to single-word descriptions.
- `Date and time` have to **strictly** be in the format as stated above.

Example:

- `range study for exam /from 01-10-2019 21:00 /to 05-10-2019 11:00`

Adds a task with description and time period to `study for exam` and between `01-10-2019 21:00` to `05-10-2019 11:00`.

3.9.6. Viewing two different calendars side-by-side: `viewDual`

Transforms all current tasks into two calendar views, one for tutor tasks and one for student tasks.

Format: `viewDual`

3.9.7. Exporting the calendar: `export`

You can export all current tasks recorded into a shareable format in calendar view.

Format: `export`

4. FAQ

Q: How do I view my tasks on the Calendar?
A: You can use the `calendar` command.

Q: How do I transfer my data to another Computer?
A: You can copy the entire directory containing our program into the destination directory.

Q: How do I save my tasks in **OOF**?
A: You are not needed to explicitly save the tasks as **OOF** will automatically save all tasks that are added during runtime.

5. Command Summary

5.1. Available Commands

View the list of features and their usages.

- **Help**: `help`

> You can view the usage of a specific command by typing `help COMMAND`, where `COMMAND` is the name of the feature. e.g. `help calendar`

Add a task with a deadline.

- **Deadline**: `deadline DESCRIPTION /by DD-MM-YYYY HH:MM`
  e.g. `deadline homework /by 20-09-2019 13:00`

Add an event with start and end time.

- **Event**: `event DESCRIPTION /from DD-MM-YYYY HH:MM /to DD-MM-YYYY HH:MM`
  e.g. `event project meeting /from 20-09-2019 13:00 /to 20-09-2019 17:00`

Add a todo with a specific date.

- **Todo**: `todo DESCRIPTION /on DD-MM-YYYY`
  e.g. `todo withdraw money /on 19-09-2019`

Set a recurring task.

- **Recurring**: `recurring INDEX NUMBER_OF_OCCURRENCES FREQUENCY`
  e.g. `recurring 1 1 1`

List all the task you have saved in **OOF**

- **List**: `list`

Mark a task as done.

- **Done**: `done INDEX`
  e.g. `done 1`

Delete a specific task.

- **Delete**: `delete INDEX`
  e.g. `delete 1`

Find anything using keywords.

- **Find**: `find DESCRIPTION`
  e.g. `find withdraw money`

Set a threshold in hours for reminders.

- **Threshold**: `threshold HH`
  e.g. `threshold 48`

Check your schedule on a particular day.

- **Schedule**: `schedule DD-MM-YYYY`
  e.g. `schedule 04-10-2019`

View a summary of your tasks for the next day.

- **Summary**: `summary`

View free time slots on a specific day.

- **Free**: `free DD-MM-YYYY`
  e.g. `free 10-10-2019`

View all tasks in a table form for any particular week.

- **ViewWeek**: `viewweek DD MM YYYY` e.g. `viewweek 30 10 2019` ℹ Note that the parameters `DD MM YYYY` are optional and the command will automatically show tasks for the current week if these parameters are not shown. The tasks in each day is chronologically sorted.

View tasks for any particular month in calendar format.

- **Calendar**: `calendar MM YYYY`
  e.g. `calendar 10 2019`

ℹ Note that the parameters `DD MM YYYY` are optional and the command will automatically show tasks for the current month if these parameters are not shown. The tasks in each day is chronologically sorted.

View reminder based on the threshold set.

- **Reminder**: `NIL`

ℹ This feature runs in the background thus no input is needed from you.

Starts Task tracker.

- **Start Task Tracker**: `tracker /start TASK_INDEX MODULE_CODE`
  e.g. `tracker /start 20 cs2113t`

Pauses Task tracker.

- **Pause Task Tracker**: `tracker /pause TASK_INDEX MODULE_CODE`
  e.g. `tracker /pause 20 cs2113t`

Stops Task tracker.

- **Stop Assignment Tracker**: `tracker /stop TASK_INDEX MODULE_CODE`
  e.g. `tracker /stop 20 cs2113t`

View Task tracker diagram.

- **View Task Tracker**: `tracker /view TIME_PERIOD`
  e.g. `tracker /view week`
- **List Task Trackers**: `tracker /list`
- **Delete a Task Tracker**: `tracker /delete TRACKER_INDEX`
  e.g. `tracker /delete 1`

Exit **OOF** by using this command.

- **Bye**: `bye`

5.2. Coming Soon

- **Tentative**: `tentative DESCRIPTION`

- **Do-after**: `Do-after INDEX DESCRIPTION`
- **Filter**: `filter CATEGORY`
- **ViewUndone**: `viewUndone`
- **Range**: `range`
- **ViewDual**: `viewDual`
- **Export**: `export`