

# Le Duc - User Guide

1. Introduction .....	1
2. Quick Start .....	2
3. Features .....	2
3.1. Viewing all the command: <code>help</code> .....	3
3.2. Adding a todo task: .....	3
3.3. Adding a deadline task: .....	3
3.4. Adding a event task: .....	4
3.5. Find the task by keyword : <code>find</code> .....	4
3.6. Delete a task : <code>delete</code> .....	4
3.7. Show all the tasks: <code>list</code> .....	5
3.8. Mark a task as done : <code>done</code> .....	5
3.9. Snooze a deadline task : <code>snooze</code> .....	5
3.10. Postpone a deadline task : <code>postpone</code> .....	5
3.11. Reschedule an event task : <code>reschedule</code> .....	5
3.12. Reminder for upcoming tasks : <code>remind</code> .....	5
3.13. Edit a task : <code>edit</code> .....	6
3.14. Customize the welcome message : <code>setwelcome</code> .....	6
3.15. Get some statistics .....	7
3.16. Sort by .....	7
3.17. Prioritize .....	7
3.18. Display by day/week/month .....	8
3.19. Shortcut .....	8
3.20. View Unfinished tasks .....	8
3.21. Create task list (not implemented yet) .....	9
3.22. Display different task list (not implemented yet) .....	9
3.23. Import other task list (not implemented yet) .....	9
3.24. Subtask (not implemented yet) .....	9
4. FAQ .....	10
5. Command Summary .....	10

By: `[AY1920S1-CS2113-T16-1]` Le Duc

## 1. Introduction

Le Duc ("The Duke" in French) is for students who have both academic and extracurricular lives and have to manage them both. Outside, they have to go to school, work part-time, attend club and have fun at some party. At home, they have to do homework, undertake a project, do some chores, cook some nice food. Pretty busy, right? With this new tool, a student can manage and organise his time more efficiently.

## 2. Quick Start

1. Please ensure that you have Java **11** or above installed.
2. Download the latest leduc jar file.
3. Execute the jar file.
4. Type the command in the shell.
5. Please refer to [Section 3, “Features”](#) part for details of each command.

A screenshot of a terminal window showing the 'leduc' application. At the top left is a logo made of characters. Below it, the application greets the user: 'Hello I'm Duke' and 'What can I do for you ?'. A horizontal line separates this from the 'help' section. The word 'help' is written in green. Below it, another horizontal line is followed by a detailed list of commands and their parameters. The commands include: 'todo', 'deadline', 'event', 'list', 'bye', 'done', 'delete', 'find', 'snooze', 'postpone', 'reschedule', 'remind', and 'help'. Each command is followed by a brief description of its function and any required parameters like DESCRIPTION, DATE, INDEX, or KEYWORD. The terminal has a dark background with light-colored text.

```
leduc

Hello I'm Duke
What can I do for you ?

help

All command will be display as :
commandName [PARAMETERS] : description of the command
All parameters will be written in UPPER_CASE
Parameters are :
DESCRIPTION : the description of a task
DATE : the date of a task
INDEX : the index of the task (goes from 1 to ...)
KEYWORD : the keyword to find a task
Date format is DD/MM/YYYY HH:mm
All blank space should be respected
Here are the list of all command:
todo DESCRIPTION : create a todo task
deadline DESCRIPTION /by DATE : create a deadline task
event DESCRIPTION /at DATE - DATE : create an event task
list : show all the tasks
bye : exit the application
done INDEX : mark as done the task of index INDEX
delete INDEX : delete the task of index INDEX
find KEYWORD : find the task with a keyword
snooze INDEX : snooze a task of index INDEX
postpone INDEX /by DATE : postpone a deadline task
reschedule INDEX /at DATE - DATE : reschedule an event task
remind : remind the first three task
help : show the list of all command
```

## 3. Features

## Command Format

- All whitespaces must be respected.
- Words in UPPER\_CASE are parameters the users have to write:
  - DATE : The date format must be dd/MM/yyyy hh:mm where dd is the days, MM is the months, yyyy is the years, hh is the hours and mm is the minutes.
  - DESCRIPTION: The description of the task
  - WELCOME: the welcome message
  - INDEX is the index of the task (goes from 1 to the number of tasks). You can find the index of the task by showing all the task with the command list of by searching the task with the command find.
  - KEYWORD: the keyword to find a task.

### 3.1. Viewing all the command: `help`

To show the list of all command : `help`

Example:

- `help`

### 3.2. Adding a todo task:

To add a todo task : `todo DESCRIPTION` An option could be added to give a priority to the task : `todo DESCRIPTION prio INDEX`

Be careful:

- The second INDEX cannot be less than 0 or greater than 9.
- 0 is the less urgent, 9 is the most urgent

Examples:

- `todo read book`
- `todo buy book`
- `todo sell book prio 3`

### 3.3. Adding a deadline task:

To add a deadline task : `deadline DESCRIPTION /by DATE` An option could be added to give a priority to the task : `deadline DESCRIPTION /by DATE prio INDEX`

Be careful:

- The second INDEX can't be less than 0 or greater than 9.
- 0 is the less urgent, 9 is the most urgent

Example:

- `deadline CS4239: lab3 /by 21/09/2019 23:59`
- `deadline CS4239: lab4 /by 25/09/2019 23:59 prio 7`

### 3.4. Adding a event task:

To add an event task : `event DESCRIPTION /at DATE - DATE` An option could be add to give a priority to the task : `event DESCRIPTION /at DATE - DATE prio INDEX`

Be careful:

- The second INDEX can't be less than 0 or greater than 9.
- 0 is the less urgent, 9 is the most urgent

Example:

- `event CS4211: mid-term /at 07/10/2019 09:00 - 07/10/2019 11:00`
- `event CS4211: final exam /at 07/12/2019 09:00 - 07/12/2019 11:00 prio 8`

Be careful : when creating an event, two dates can't clash

### 3.5. Find the task by keyword : `find`

To find a task by keyword : `find KEYWORD`

The find command allows the user to search for tasks by keyword. The command will return a list of tasks, ordered by ascending relevancy. Typos in the user query will not affect performance.

Examples:

- `find book`
- `find buy book`

### 3.6. Delete a task : `delete`

To delete a task : `delete INDEX`

Example:

- `delete 1`

### 3.7. Show all the tasks: **list**

To show all the tasks: **list**

Example:

- **list**

### 3.8. Mark a task as done : **done**

To mark a task as done: **done INDEX**

Example:

- **done 2**

### 3.9. Snooze a deadline task : **snooze**

To snooze a deadline task: **snooze INDEX** The snooze time is fixed at 30 minutes.

Example:

- **snooze 2**

### 3.10. Postpone a deadline task : **postpone**

To postpone a deadline task: **postpone INDEX /by DATE**.

DATE is the new date of the deadline task. The new date should be after the old one.

Example:

- **postpone 2 /by 12/12/2020 22:10**

### 3.11. Reschedule an event task : **reschedule**

To reschedule an event task: **reschedule INDEX /at DATE - DATE**.

Be careful : when rescheduling an event, two dates can't clash

Example:

- **reschedule 3 /at 12/12/2020 10:00 - 12/12/2020 12:00**

### 3.12. Reminder for upcoming tasks : **remind**

Reminds the user of the 3 most upcoming tasks : **remind**

Example:

- `remind`

### 3.13. Edit a task : `edit`

To edit a task: `edit`

To edit a task, follow these instructions:

1. All of the tasks will be displayed, you have to choose a task INDEX
2. Depending on the type of task:
  - If it is a todo task, you can enter the new DESCRIPTION of the task
  - If it is not a todo task, you have to choose 1) if you want to edit the description or 2) if you want to edit the date
    - Then, enter the new DESCRIPTION or the new DATE of the task

Examples:

- Edit the description of the task 2 ( todo task )
  - `edit`
  - `2`
  - `DESCRIPTION`
- The task 2 is a deadline task:
  - `edit`
  - `2`
  - `2`
  - `DATE`

### 3.14. Customize the welcome message : `setwelcome`

To customize the welcome message: `setwelcome WELCOME`

Example:

- `setwelcome hello World`

Be careful:

\*Reverting to the previous welcome message is not possible once a new message is set. \*Ensure the directory data is in the correct location === Exit the program : `bye`

To exit the program : `bye`

Example :

- `bye`

## 3.15. Get some statistics

Display statistics : `stats`

Example:

- `stats`

Output:

Here are some statistics about your task list:

Number of tasks: 13.0

Number of Todo's : 6.0

Number of Events: 2.0

Number of Deadlines: 5.0

Number of Uncompleted Tasks: 13.0

Number of Completed Tasks: 0.0

Percent Complete: 0.0%

## 3.16. Sort by

Sort all task by date/description: `sort SORTTYPE`

SORTTYPE is either date, description, priority

Be careful:

- Sorting "by date" will sort tasks in chronological order
- Sorting "by description" will sort the descriptions in alphabetical order
- Sorting "by priority" will sort tasks in ascending urgency

Examples:

- `sort date`
- `sort priority`
- `sort description`

## 3.17. Prioritize

Giving priority to task: `prioritize INDEX prio INDEX`

The first INDEX is the task index

The second INDEX is the priority (goes from 0 to 9)

Be careful:

- The second INDEX can't be less than 0 nor greater than 9.
- 0 is the less urgent, 9 is the most urgent
- When creating a task, specifying the priority is optional. When the priority is not specified, the task will automatically have a priority of 5.

Example:

- `prioritize 4 prio 2`
- `prioritize 5 prio 8`

## 3.18. Display by day/week/month

Display all the task for one particular day/week/month: `display date DATETYPE DATE`

DATETYPE is day, week, month.

DATE is

- a date for the DATETYPE day
- the first date of the week for the DATETYPE week
- a month and the year for the DATETYPE month

Be careful:

- DATE must be in dd/MM/yyyy format for the DATETYPE day and week
- DATE must be in MM/yyyy format for the DATETYPE month

## 3.19. Shortcut

Give shortcut to command : `shortcut`

For each command, the console will output the command name.

The user (you) have to input the shortcut you want.

If the shortcut already exists, the console will tell you to assign another shortcut

## 3.20. View Unfinished tasks

Find and display all unfinished tasks

Output:



Here are the unfinished tasks in your list:

1. [T][ ] td1 [Priority: 5]
2. [D][ ] d1 by: 14/09/2019 22:33 [Priority: 5]
3. [E][ ] e1 at: 21/09/2019 00:00 - 28/10/2019 22:22 [Priority: 5]
4. [T][ ] td4 [Priority: 5]
5. [T][ ] td5 [Priority: 5]

-----

## 3.21. Create task list (not implemented yet)

Create a different task list: `tasklist DESCRIPTION`

DESCRIPTION is the name of the new tasklist

Be careful:

- The first tasklist will be name main task list, except if it is edited (can't be done for the moment)
- Each new tasklist will have an index automatically

## 3.22. Display different task list (not implemented yet)

Display different task list, for example one task list for school, one task list for work: `display tasklist INDEX`.

INDEX is the task list index (if exist)

## 3.23. Import other task list (not implemented yet)

If the user want to import another task list: `Import FILENAME`.

The FILENAME is path to the file

Be careful:

- The file must be in the same format as every file in the project
- The file must be in the directory data

## 3.24. Subtask (not implemented yet)

You can create a subtask of another task: when creating the task, just add `sub INDEX` at the end.

INDEX is the index of the task

## 4. FAQ

Q) Will there be more feature ?

A) Yes, if we have enough imagination and time and energy.

## 5. Command Summary

- **help** : `help`
- **todo** : `todo DESCRIPTION`
- **deadline** : `deadline DESCRIPTION /by DATE`
- **event** : `event DESCRIPTION /at DATE - DATE`
- **find** : `find KEYWORD`
- **delete** : `delete INDEX`
- **list** : `list`
- **done** : `done INDEX`
- **snooze** : `snooze INDEX`
- **postpone** : `postpone INDEX /by DATE`
- **reschedule** : `reschedule INDEX /at DATE - DATE`
- **remind** : `remind`
- **edit** : `edit` and follow the instructions
- **setwelcome** : `setwelcome WELCOME`
- **bye** : `bye`
- **stats** : `stats`
- **sort** : `sort SORTTYPE`
- **prioritize** : `prioritize INDEX prio INDEX`
- **display** : `display date DATETYPE DATE`
- **shortcut** : `shortcut`

### Not implemented yet

- **create a tasklist** : `tasklist DESCRIPTION`
- **display tasklist** : `display tasklist INDEX`
- **import file** : `Import FILENAME`
- **subtask** : when creating the task, just add `sub INDEX` at the end