Diary (of the) Undergraduate Kommon Engineer (D.U.K.E.) Manager - User Guide

By: Team AY1920S1-CS2113-T13-4 Since: Aug 2019 Licence: MIT

1. Introduction **2**

2. Quick Start **2**

3. Features **2**

3.1. Task Creation 3

3.1.1. Creating a new "To-Do" task : todo 3

3.1.2. Creating a new "Deadline" task : deadline 3

3.1.3. Creating a new "FixedDuration" task : fixedduration 3

3.2. Expenditure Tracker 4

3.2.1. Create a new budget : new 4

3.2.2. Deducts expenses to budget : minus 4

3.2.3. Add earnings to budget : add 5

3.2.4. View currently available budget : view 5

3.2.5. Resets budget : reset 5

3.2.6. Undoes budget : undo 5

3.3. Contacts 6

3.3.1 Adding a contact : addcontact 6

3.3.2. Listing all contacts : listcontacts 6

3.3.3 Find contacts : findcontact 6

3.3.4 Delete specific contacts : deletecontact 7

3.4. Notes 7

3.4.1 Notes in a task : notes 7

3.5. Priority 8

3.5.1. Set the priority of a task: setpriority 8

3.5.2. Browse the list of tasks and their priorities: priority 8

3.5.3. Find a list of tasks that have a certain priority: findpriority 8

3.6. Generic Task Management 9

3.6.1. Listing all tasks : list 9

3.6.2. Finding particular tasks by task description : find 9

3.6.3. Marking a completed task : done 9

3.6.4. Deleting a task : delete 10

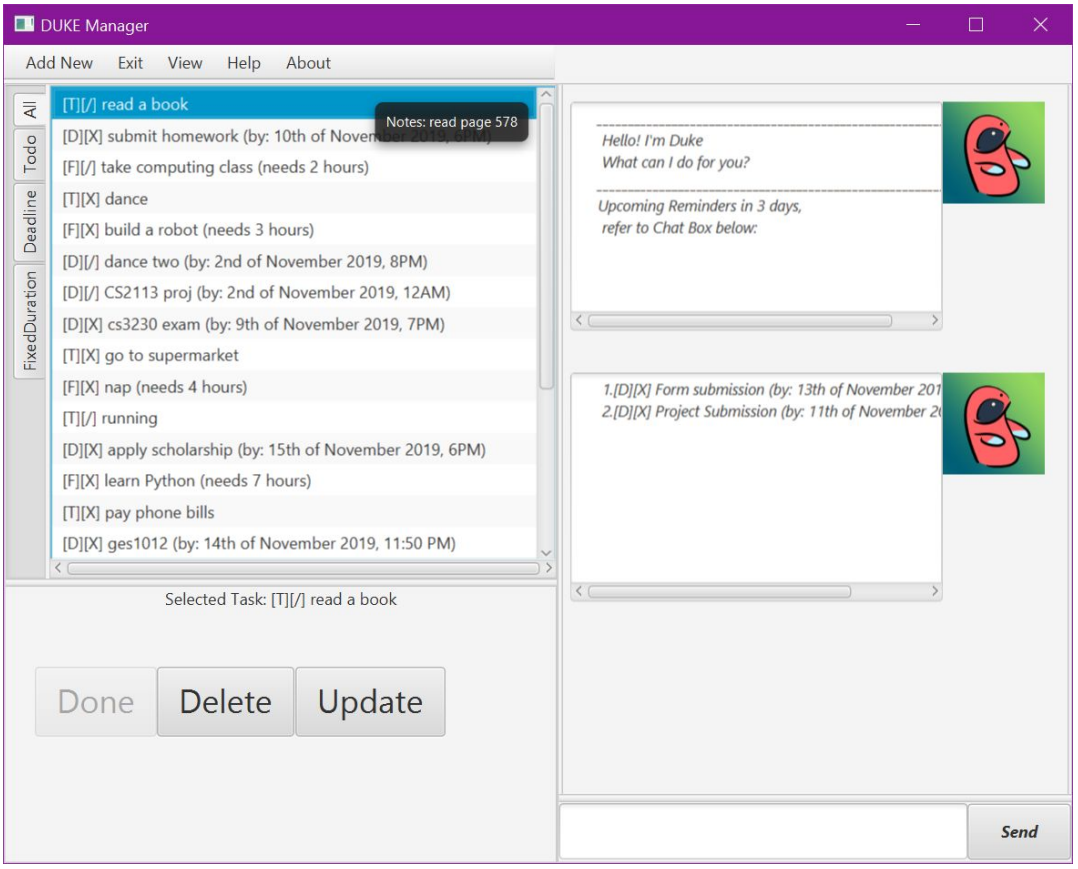
3.6.5. Updates an existing task in tasklist: update 10
3.6.6. Detect duplicates in task list 11
3.6.7. Detecting anomalies in task list 11
3.6.8 View all available functions or commands: help 11
3.6.9 Filters tasks that are of the same type in tasklist: filter 11
3.6.10. Find a list of tasks on a certain date: finddate 12
3.6.11. Retrieving upcoming reminders of tasks due in 3 days 12
3.7. Miscellaneous Inputs 12
3.7.1. Backing up the program : backup 12
3.7.2. Exiting the program : exit 13
3.7.3. Saving the data 13
3.7.4 Friendlier Syntax 13
3.8. On-App Guide 13
3.8.1. Opens the interface guide : help 13
4. Additional Features (Coming in v2.0) **14**
4.1. Academic Calendar 14
4.2. Module Structure 14
4.3. History 14
5. FAQs **14**
6. Command Summary **14**

1. Introduction

DUKE Manager is for those who prefer an electronic handbook as compared to a hard copy one. It is also optimised for those who prefer to input notes as a CLI / text messenger style.

2. Quick Start

1. Ensure you have Java 11 or above installed in your Computer.
2. Clone the Github to your computer and launch the GUI stored in main/src/main/java/duke/Launcher.java
3. Alternatively, you may download the jar file under releases.
4. If there are no existing data files, Duke Manager will create empty files to store data.



5. Type a command in the text box and press “Send” or hit the “Enter” key on your keyboard to execute it.
6. Some example commands you can try:
    - `todo housework` : adds “housework” into list of todo tasks
    - `list` : lists all tasks
    - `delete 1` : deletes the first task stored in the list of tasks
    - `exit` : exits the app

3. Features

Command Format

- Words enclosed in < > are the parameters to be supplied by the user e.g. in todo <task description>, <task description> is a parameter which can be used as todo homework.

3.1. Task Creation

3.1.1. Creating a new "To-Do" task : todo

Creates a new task called "To-do" in Duke Manager.

Example:
Let's say you would like to add a task to be done later such as an assignment from school.

- todo homework
- todo exercise after school
  All these adds into a list of tasks to be done.

Format: `todo <task description>`
- The task description can be any work you are planning to complete.

3.1.2. Creating a new "Deadline" task : deadline

Creates a new task called "Deadline" in Duke Manager. It is used to allow the user to take notice of the date and time of the task's deadline as an additional piece of information.

Example:
Let's say you have a task that needs to be completed before a specific time. Thus, you would like to input the date and time.

- deadline finish quiz /by 11/11/2019 2359
  This adds to the list of deadlines to show when it is supposed to be completed.

Format: `deadline <task description> /by <date and time>`
- Date and time have to be inputted in this format <dd/MM/yyyy HHmm>.

3.1.3. Creating a new "FixedDuration" task : fixedduration

Creates a new task called "FixedDuration" in Duke Manager.
Fixed-duration tasks are defined to be tasks that do not have a specific deadline but time required/allocated to finish these tasks is fixed. (Usually these tasks last for a relatively long time, i.e. longer than 10 minutes)

Example:
Let's say you want to add a task that has no deadline, but you know approximately how long you are going to be doing them. Thus, you add in a task that states the duration which the task will take.

- workout for 1 hour
- running for 30 minutes
- reading a book for 2 hours
  These adds to the list of tasks with fixed duration.

Format: `fixedduration <task description> /for <duration> <unit>`
- The duration refers to the length of time.
- The unit refers to the time unit and can be either in hours or minutes.

3.2. Expenditure Tracker

This is a simple expenditure tracker that users can use to track their expenses in school. The initial budget to be entered by the user can either be their initial budget (amount in the bank, or monthly allowance), or the limits they set for themselves (Spending only $250 per month, etc)

The budget tracker commands starts with “budget”.

3.2.1. Create a new budget : new

Creates a new budget if there is no budget created. Upon creation of the new budget, any preexisting budget will be cleared.

Example:
Let’s say you want to create a new budget for the new school term.

- budget new 1000

Format: budget new <amount>

- The amount refers to the amount of money that you either currently have, or am limiting yourself to spend.
- Amount is limited to between -999,999 and 999,999.
- All previous entries will be cleared upon entering this command.

3.2.2. Deducts expenses to budget : minus

Deducts the expenses from current available budget, with an optional description.

If the user does not input any description, it will input "No Description" instead.

Example:
Let’s say you spent some money to purchase for an E-scooter.

- budget minus 999 Bought an E-scooter

Format: budget minus <amount> <(Optional)Description>

- Amount refers to the amount spent.
- Amount is limited to between -999,999 and 999,999.
- Description is optional; it acts like a note on what is the amount spent on.
- If there are no descriptions detected, it will input the remarks as “No Description”.
- The command “minus” can be replaced with a “-”. The spacing between - and <amount> should still be there.
    - Eg: budget - 999 Bought an E-scooter

3.2.3. Add earnings to budget : add

Adds the earnings to the current available budget, with an optional description.

Example:
Let's say you spent some money to purchase for an E-scooter.

- budget add 400 Sold an E-scooter

Format: budget add <amount> <(Optional)Description>

- Amount refers to the amount earned or obtained.
- Amount is limited to between -999,999 and 999,999.
- Description is optional; it acts like a note on what is the amount spent on.
- If there are no descriptions detected, it will input the remarks as "No Description".
- The command "add" can be replaced with a "+". The spacing between + and <amount> should still be there.
    - Eg: budget + 400 Sold an E-scooter

3.2.4. View currently available budget : view

Shows the user the current available budget, as well as the total earnings and expenses recorded.

Format: budget view

3.2.5. Resets budget : reset

Resets the budget list with the initial input being the one that is defined. The existing list will be cleared as well.

Example:
Let's say you want to reset your monthly allowance for the new year.

- budget reset 1000
  Resets the budget to $1000

Format: budget reset <amount>

- The amount refers to the amount of money that you either currently have, or am limiting yourself to spend.
- Amount is limited to between -999,999 and 999,999.
- All previous entries will be cleared upon entering this command.

3.2.6. Undoes budget : undo

Undoes the latest entry in the budget list and updates the budget accordingly.

Format: budget undo

3.3. Contacts

3.3.1 Adding a contact : addcontact

This command allows the user to add a new contact that stores name, number, email and office. If any of the details are unavailable, simply leave a blank space.

Example:
Let's say you have the tendency to drop your phone and are concerned that one day you might damage your phone and it can never power on again, losing precious data especially a professor's contact number. Thus, you decided to store important contact information in DUKE Manager instead.

- addcontact Christian, 89974554, christian@u.nus.edu, E7-8-2
  This adds to the list of contacts.

Format: `addcontact <name>, <number>, <email>, <office>`

- <name> refers to the name of the contact to be added.
- <number>, <email> and <office> of the contact is to be input in this format.
- <email> must contain an "@" to qualify as an appropriate address.
- For details that are not known, simply omit it.

3.3.2. Listing all contacts : listcontacts

This command shows the user all the contacts that have been saved.

Example:
Let's say you want to see all the contacts saved thus far and double check to see what you entered is accurate.

Format: `listcontacts`

3.3.3 Find contacts : findcontact

This command finds and displays relevant contacts stored inside DUKE Manager.

Example:
Let's say you have stored a lot of contact and would like to quickly find a specific person's details, but you only remembered part of the person's name. Thus, you find by inputting that detail and DUKE Manager will find it for you. This also works for numbers, emails and office location.

- findcontact Tan
  Finds the list of contacts to search for any input with "Tan", regardless of it being upper or lower case.
- fc 91237978
  Shortcut to search through the contact list for this phone number and displays it accordingly.

Format: `findcontact <keyword>`

- <keyword> is not case sensitive.
- Can search for name, number, email or office.
- Partial words will be matched. Eg. Tan will match Prof Tan.
- Only contacts will be searched.

3.3.4 Delete specific contacts : deletecontact

This command allows the user to delete a specific contact inside DUKE Manager.

Example:
Let's say the contact details are outdated and you would like to remove it totally from the system since you do not have that person's new contact details.

Format: deletecontact <index>

- Deletes the contact at the specified <index>.
- The index refers to the index number shown in the displayed list contact.
- The index must be between 1 and the total number of contacts in the contact list.

3.4. Notes

3.4.1 Notes in a task : notes
Notes can be either added or updated to an existing task in the tasklist. They can also be removed from an existing task. Notes of an existing task can be shown in the tasklist.

Example:
Let's say you would like to add notes to the first task in the list so as to give a better description on what to do and view it later. After the note is not useful, you will also want to remove it to keep a clean diary.

- notes 1 /add read page 578
  Adds/Updates notes of the first task in the tasklist.

- notes 2 /show
  Shows notes of the second task to the user in the tasklist.

- notes 3 /delete
  Deletes notes of the third task in the tasklist.

Format: notes <task number> <type of notes> <(Optional) notes description>

- Adds/Updates/Deletes/Shows notes of the task at the specified <task number>.
- The task number refers to the index number shown in the displayed task list.
- The index **must be in between 1 and the size of tasklist**. e.g. For a tasklist that contains 4 tasks, only numbers 1 to 4 are allowed.
- The type of notes refers to either /add, /delete, or /show.
- /add represents adding/updating notes of the task.
- /delete represents removing existing notes from the task.
- /show represents showing the notes of the task.

- The notes description is **only required when adding or updating notes**, not when deleting or showing notes.

3.5. Priority

Priority is a user-defined attribute for each task in the task list. It represents the urgentness/importance of a certain task.

Priority has 5 levels, with each represented an integer number. The number ranges from 1, the highest, to 5, the lowest priority.

The priority of a task defaults to the lowest, level 5, initially when it is created.

Users can set priority, find tasks by priority, browse tasks with priority using the following commands:

3.5.1. Set the priority of a task: setpriority

Change the default priority of a task to a user-set priority.

Example:
Let's say after adding a few tasks, you realise you want to shift the position of some tasks as they have higher priority than others. Thus, you set the priority of the task using this function.

- setpriority 4 1
  Set task number 4 to a high priority.

- setpriority 5 3
  Set task number 5 to a mid range priority.

Format: `setpriority <task number> <priority>`

- Set the priority of the task at the specified <task number>.
- The <task number> refers to the index number shown in the displayed task list.
- The index must be in between 1 and the size of tasklist. e.g. For a tasklist that contains 4 tasks, only numbers 1 to 4 are allowed.
- The <priority> refers to the priority of the task: 1-HighPriority ~ 5-LowPriority.
- The priority must be between 1 and 5 both inclusive.
- Alerts user that the priority has been changed.

3.5.2. Browse the list of tasks and their priorities: priority

Display the list of tasks and their priorities beside each task.
(Tasks are sorted in descending order of their priorities)

Format: `priority`

3.5.3. Find a list of tasks that have a certain priority: findpriority

Return a list of tasks that have a certain priority.

Example:
Let's say after setting quite a few tasks to have different priorities, you would like to consolidate all the tasks with high priority.

- findpriority 1

Format: `findpriority <priority>`

- The <priority> refers to the priority of the task: 1-HighPriority ~ 5-LowPriority.
- The priority must be between 1 and 5 both inclusive.
- If no records are found, the user will be alerted.

3.6. Generic Task Management

3.6.1. Listing all tasks : list

Shows a list of all tasks in Duke Manager.

Format: `list`

3.6.2. Finding particular tasks by task description : find

Finds tasks inside the tasklist and returns the list of tasks found.

Example:
Let's say you would like to find a specific task in a long list of tasks. To quickly find it, simply use this function.

- find homework
  This will search through the tasks for a description that contains the word homework.

Format: `find <keyword>`

- The search is case sensitive. e.g. run will not match Run
- Partial words will be matched. e.g. ru will match run
- Only task description is searched.

3.6.3. Marking a completed task : done

Labels a task with the specified index as done.

Example:
Let's say you would like to mark a completed task as done.

- done 1
  This mark task one as done with a tick.

Format: `done <task number>`

- Marks the task at the specified <task number>.
- The task number refers to the index number shown in the displayed task list.

- The index **must be in between 1 and the size of tasklist**. e.g. For a tasklist that contains 4 tasks, only numbers 1 to 4 are allowed.

3.6.4. Deleting a task : delete

Deletes the specified task from the tasklist.

Example:
Let's say a task is completed and may be removed.
- delete 1
  This removes the first task from the list.

Format: `delete <task number>`
- Deletes the task at the specified <task number>.
- The task number refers to the index number shown in the displayed task list.
- The index **must be in between 1 and the size of tasklist**. e.g. For a tasklist that contains 4 tasks, only numbers 1 to 4 are allowed.

3.6.5. Updates an existing task in tasklist: update

Updates the task, either task description, date and time, or type of task in Duke Manager.

Example:
Let's say there is an update about a certain task, and you would like to change what was stored in the task without deleting and adding again.
- update 1 /desc homework
  Updates the 1st task description in the tasklist.

- update 5 /date 17/09/2019 1222
  Updates date and time of the 5th task in the tasklist.

- update 2 /type deadline
  Updates type of the 2nd task to Deadline in the tasklist.

Format: `update <task number> <type of update> <description to be updated>`
- Updates the task at the specified <task number>.
- The task number refers to the index number shown in the displayed task list.
- The index **must be in between 1 and the size of tasklist**. e.g. For a tasklist that contains 4 tasks, only numbers 1 to 4 are allowed.
- The type of update refers to either /desc, /date, or /type.
- /desc represents updating the task description.
- /date represents updating the date and time of the task.
- /type represents updating the type of task.
- The description to be updated refers to either description of task, date and time, or type of task depending on <type of update>.
- Date and time have to be inputted in this format <dd/MM/yyyy HHmm>.
- Returns an error if a task does not contain date and time when the user tries to update date and time of the particular task.

- Returns an error if the task type is invalid.

3.6.6. Detect duplicates in task list

This feature finds and alerts the user of duplicated tasks which may be re-entered. This works for Todo, Deadline and FixedDuration only.

Example:
Let's say you have entered a task to do homework but after keying in other tasks, you have forgotten that homework was already entered and thus, you enter the same task again. Without this feature, when you mark the first homework task as done, you may get confused as to why it still appears in the list of undone tasks. Thus, detect duplicates prevents that from occurring.

- todo MA1508E homework
  The same task is already in the list!

3.6.7. Detecting anomalies in task list

Detects tasks that clash with the same date and time when adding a new task, or updating an existing task in Duke Manager.

- Alerts user that an existing task has the same date and time.

- Prompts the user to either pick a different date and time, or mark the existing task as done.

Example:

- deadline homework /by 17/09/2019 1222
  Returns (>_<) OOPS!!! The date/time for deadline clashes with <E><X> concert (at: 17th of September 2019, 12:22 PM) Please choose another date/time! Or mark the above task as Done first!

3.6.8 View all available functions or commands: help

View all the functions and commands that Duke Manager have.
Format: help

3.6.9 Filters tasks that are of the same type in tasklist: filter

Categorizes tasks in the task list based on the type of task.
Examples:
- filter todo : Returns a list of Todo tasks.

- filter deadline : Returns a list of Deadline tasks.

Format: filter <type of task>
- The type must be in lowercase. e.g. todo is not Todo
- Returns an error if the task type is invalid.

3.6.10. Find a list of tasks on a certain date: finddate

Return a list of tasks due on a certain date.
Format: finddate /on <date>

- The <date> is in the format <dd/MM/yyyy>.
- If no records are found, the user will be alerted.

3.6.11. Retrieving upcoming reminders of tasks due in 3 days

Upon proper exiting of Duke Manager through the exit command, the progress of all the data will be saved. Next, open/double click the Jar File once again, this returns back to the program after exiting for the first time previously.

This will then prompt Duke Manager to load the saved progress of the inputs you have keyed in earlier as existing tasks. Additionally, the interface on the right will retrieve and display the upcoming reminders of tasks due in 3 days from the current date today.

Does not require any additional input unless it does not meet prerequisites.

Prerequisites:

- Ensure that there is an existing Priority-Related Task Input (task with a time and date) added previously.

  Example: deadline <task description> /by <date and time>

- Ensure that the existing Priority-Related Task added previously <date and time> is set to between 1-3 days after <todaysDate>.

  Example: Today's date is 14/11/2019. Must exist a task whereby it is already saved in the program. eg. deadline <task description> /by 15/11/2019 18:00

3.7. Miscellaneous Inputs

3.7.1. Backing up the program : backup

Backs up the current state of Duke Manager and opens the file explorer containing the data for importing / exporting.

Format: backup

3.7.2. Exiting the program : exit

Exits the program and overwrites the previous save file with the updated tasklist.
Format: exit

3.7.3. Saving the data

Duke Manager data is saved in the hard disk automatically upon exiting. There is no need to save manually. **Note: Any unsaved progress will be lost upon exiting with task manager's End Task function.**

3.7.4 Friendlier Syntax

As some commands are longer than the others, a shortened command has been provided for the ease of the user input. The table below shows the original command, and the simpler or alternative syntax.

Original Command	**Alternative/Shortened Commands**
deadline	dl
fixedduration	fd
priority	lp
setpriority	sp
findpriority	fp
addcontacts	ac
listcontacts	lc
deletecontacts	dc
findcontacts	fc
exit	bye
budget add	budget +
budget minus	budget -
budget view	budget list

3.8. On-App Guide

3.8.1. Opens the interface guide : help

Shows a friendly help tutorial on how to use DUKE Manager. This opens a new help window to allow the user to view commands via user friendly interface.

Format: `help`

4. Additional Features (Coming in v2.0)

These are the features that we have planned out for v2.0, but due to various restraints in time we could not manage to feature them.

4.1. Academic Calendar

This feature would consist a term and semester calendar which shows the events happening on the particular date and time. It can also be used to input examination and test dates, as well as project deadlines.

4.2. Module Structure

This feature would allow for the storing and viewing of modules into DUKE Manager, to aid the students in the planning of each semester and beyond.

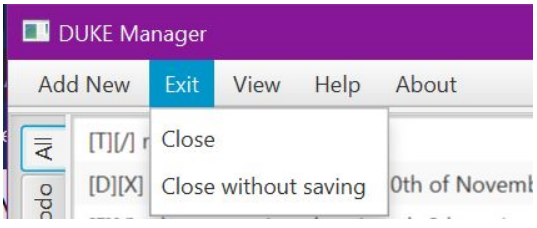
4.3. History

The user can retrieve a list of commands that were entered previously. These previous commands can also be reused again by pressing up or down arrow keys.

5. FAQs

Q: Is there an easier way to view the budgets?
A: Yes, instead of typing budget view, you can look at the menu bar at the top, under “View”, select “Budget”. This will allow you to see the budget list on a new window. Do note that if you update the budget while the window is open, you will have to refresh the budget window by re-entering to it in order to see the changes made to the list.

Q: Is there a way to exit without saving the progress?
A: Yes, instead of exiting normally, locate the menu bar at the top and click on the “Exit” command. There is an option to exit without saving.



6. Command Summary

- **Todo** : todo <TASK>
  e.g. todo CS2113 assignment
- **Deadline** : deadline <TASK> /by <DD/MM/YYYY> <HHMM>
  e.g. deadline finish reading book /by 27/10/2019 2359
- **FixedDuration** : fixedduration <TASK> /for <DURATION> <UNITS>

e.g. fixedduration build a robot /for 3 hours

- **List** : list
- **Done** : done <INDEX>
  e.g. done 1
- **Find** : find <KEYWORD> <MORE_KEYWORDS>
- e.g. find meet Christian
- **Delete** : delete <INDEX>
  e.g. delete 3
- **Add to Budget:** budget add <AMOUNT> <(OPTIONAL)DESCRIPTION>
  e.g. budget add 4.99 Friend paylah back meal money
- **Subtract from Budget:** budget minus <AMOUNT> <(OPTIONAL)DESCRIPTION>
  e.g. budget minus 21.13 Pay school fees
- **Reset Budget:** budget reset <AMOUNT>
  e.g. budget reset 100
- **View Budget:** budget view
- **Undo Previous Budget :** budget undo
- **AddContact :** addcontact <NAME>, <NUMBER>, <EMAIL>, <OFFICE>
  eg. addcontact Prof Tan, 91234567, tancc@nus.edu.sg, E1-08-11
- **ListContacts :** listcontacts
- **FindContact :** findcontact <KEYWORD>
  eg. findcontact tan
- **DeleteContact** : deletecontact <INDEX>
  eg. deletecontact 1
- **Add/Update Notes:** notes <INDEX> /add <NOTES DESCRIPTION>
  e.g. notes 1 /add read page 578
- **Show Notes:** notes <INDEX> /show
  e.g. notes 1 /show
- **Delete Notes:** notes <INDEX> /delete
  e.g. notes 1 /delete
- **Filter:** filter <TASK TYPE>
  e.g. filter deadline
- **Update task description:** update <INDEX> /desc <DESCRIPTION>
  e.g. update 1 /desc running
- **Update task date/time:** update <INDEX> /date <DATE/TIME>
  e.g. update 1 /date 27/10/2019 2359
- **Update task type:** update <INDEX> /type <TASK TYPE>
  e.g. update 1 /type todo
- **Set Priority** : setpriority <tasknum> <priority>
  e.g. setpriority 1 5
- **Find tasks based on the priority** : findpriority <PRIORITY>
  e.g. findpriority 1
- **Lists Priority in sorted order :** priority
- **Find tasks on a date** : finddate /on <DATE>
  e.g. finddate /on 01/12/2019
- **Backup Duke Manager :** backup

- **Exit Duke Manager :** exit/bye