Task scheduler project

This project is a Task scheduler, and the Reminder system command-line application implemented in C++. It allows the user to manage tasks with features such as adding, editing, and removing tasks, setting dependencies, recurring tasks, and reminders.

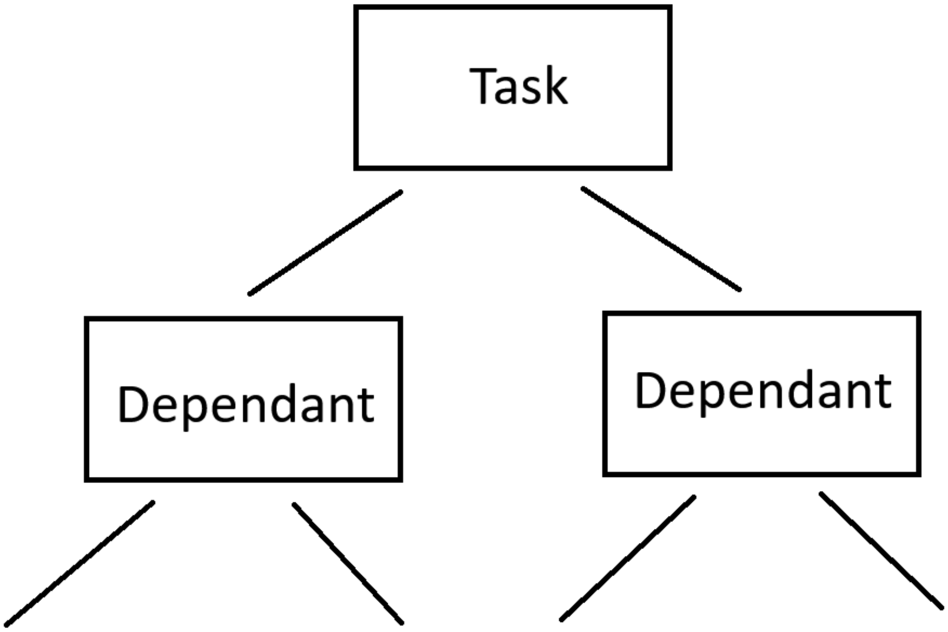
User manual

Upon running the application, the user will be greeted with a command-line interface where you can enter various commands to interact with the task scheduler.

For every command the user first enters only the command and only after they are expected to fill necessary fields. Below are the commands you can use:

- add: Command ‘add’ allows the user to add a task to the task scheduler from the command line. As the add command is given, the user is asked to provide a title, description, due date, priority, recurrence, and dependency. If the task is recurring then the user is asked to choose one of three recurrence options. If the task is dependent to other tasks, then the user must provide other tasks’ indexes. In case the user provides an invalid value in any of the fields they will be asked to retry and given instructions on how the input should be if needed.
- edit: The command ‘edit’ allows the user to edit one of the parameters in the task. As the edit command is given, the user is asked which task they would like to edit, what attribute they would like to edit, and finally to provide a new value for that attribute.
- remove: The command ‘remove’ allows the user to remove a task from the task scheduler. As the remove command is given, the user is asked to provide an index of a task they would like to be removed, and if the index is valid task is removed.
- remove all: The command ‘remove all’ allows the user to remove every task from the task scheduler using only one command.

- list: The command ‘list’ allows the user to see every task in the task scheduler. Tasks are not presented in any particular way here but task IDs do match up with their order in the list thus the list shows tasks based on their IDs in ascending order.
- listP: The command ‘listP’ allows the user to see every task based on their priority, presented in descending order (the highest priority is at the top of the list).
- show: The command ‘show’ allows the user to see all attributes of a single task. As the command show is given, the user is asked to provide an index of a task they would like to be printed.
- save: The command ‘save’ allows the user to save the current task list to the main storage file ‘tasks.txt’ (the file will be overwritten, anything that was there before will be deleted). No need to provide any input other than giving the ‘save’ command.
- completed: The command ‘completed’ allows the user to set the state of the task to complete. As the command completed is given, the user is asked to provide an index of the task they have completed. Completion of certain tasks means that its dependents have also been completed (and also dependants’ dependants and so on, so we go traverse down to the leaves). We also have to check above if completion of this certain task could also cause completion of something else as a task is automatically completed if all its dependants are completed.



- incomplete: The command ‘incomplete’ allows the user to set the state of the task to incomplete. As the command incomplete is given, the user is asked to provide an ID of a task they would like to set as incomplete. The given task is set a incomplete and so are its dependants (they are also set to incomplete). The completion status of tasks in the dependency list is also set to incomplete as one of their dependants is now incomplete thus they are also incomplete.
- export: The command ‘export’ allows the user to export the tasks to any file. As the export command is given, the user is asked to provide the name of the file in which they want to export tasks. If the file exists, the tasks are exported to it and if it does not exist the file with the provided name is created and tasks are exported to it.
- import: The command ‘import’ allows the user to import tasks from any previously exported file (the format should be correct for each task). As the command import is given, the user is asked to provide the name of the file

from which tasks will be imported, if such file exists import is successful otherwise user is notified about failure. It must be noted that the import feature deletes all tasks that are currently in the task scheduler.

- quit: The command ‘quit’ allows the user to exit the application. This is the safest and recommended way of exiting the application. This command makes sure the tasks are saved to the local file and they are not lost. If the application is closed through any other option current task list won’t be saved if the user has not called the ‘saved’ command just before the exit.
- help: The command ‘help’ provides the list of commands and their short definitions.