COMPal - User Guide

By: `AY1920S1-CS2113T-W17-1` Last Updated: `26/09/2019` License: `MIT`

Table Of Contents

1. Introduction

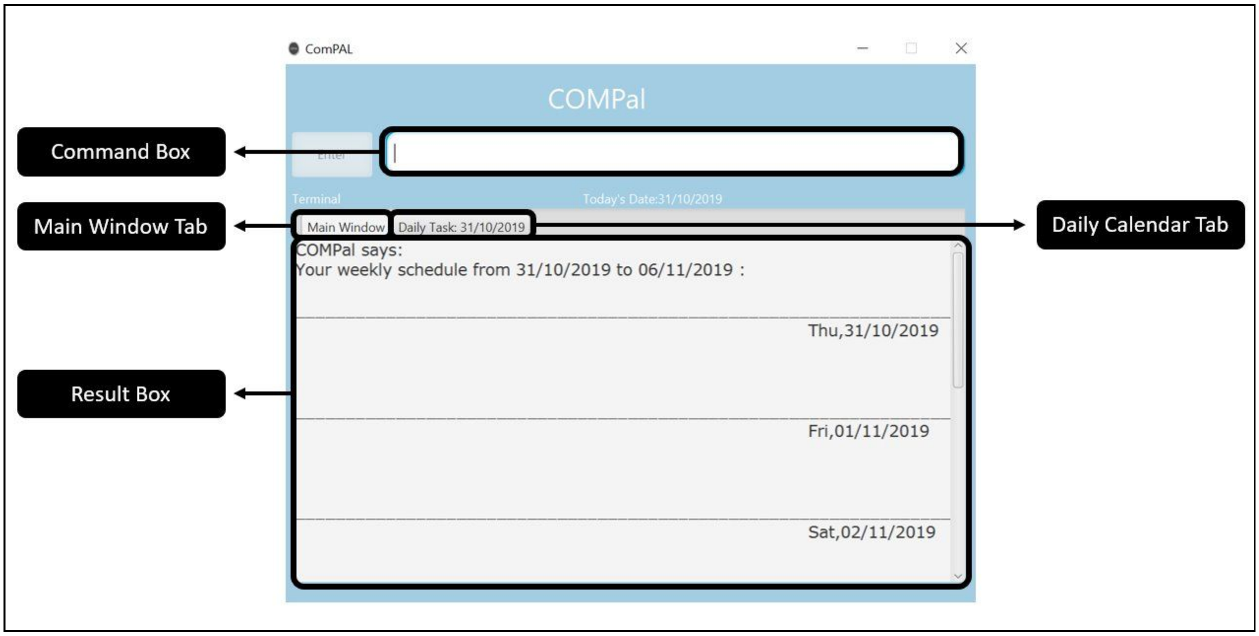
Welcome to **COMPal**!

COMPal is a Command Line Interface calendar application that targets students who prefer to use a desktop application for managing their busy student life. **COMPal** captures your timetable in a ***user-friendly layout***, giving you an ***informative overview*** of your schedule in a brief glance.

Additionally, you can ***include non-academic activities*** along with your academic timetable, unlike other widely-used timetable tools. You also have the ***freedom to prioritise certain tasks*** over less important ones and make use of ***timely reminders*** on pending **tasks**.

Take control with our **Quick Start** guide. ***COMPal*** your life, today.

The below is how COMPal interface looks like!



```
COMPal says:
Your weekly schedule from 31/10/2019 to 06/11/2019 :
______________________________________________________
                                        Thu,31/10/2019
______________________________________________________
                                        Fri,01/11/2019
______________________________________________________
                                        Sat,02/11/2019
```

Figure 1. Graphical user interface screen upon start-up

2. Understanding This User Guide

This guide explains how you can use **COMPal** to divide your time between your academic commitments and non-academic activities.

You can find comprehensive steps on how to fully utilise **COMPal**'s extensive suite of time-management tools. **Frequently Asked Questions** can also clarify any pressing doubts that you may have. Our **Command Summary** provides a concise, easy-to-read summary of our commands for your easy perusal.

You can use the **Table of Contents** above to navigate effortlessly between each section.

We have developed a list of icons below that will help you a lot in digesting our material.

Icon	**What does it mean?**
ℹ	Important information that you may want to take note of
💡	Tips and Tricks! Follow these suggestions to make your life simpler.
⚠	Warning! You need to be careful when this appears.

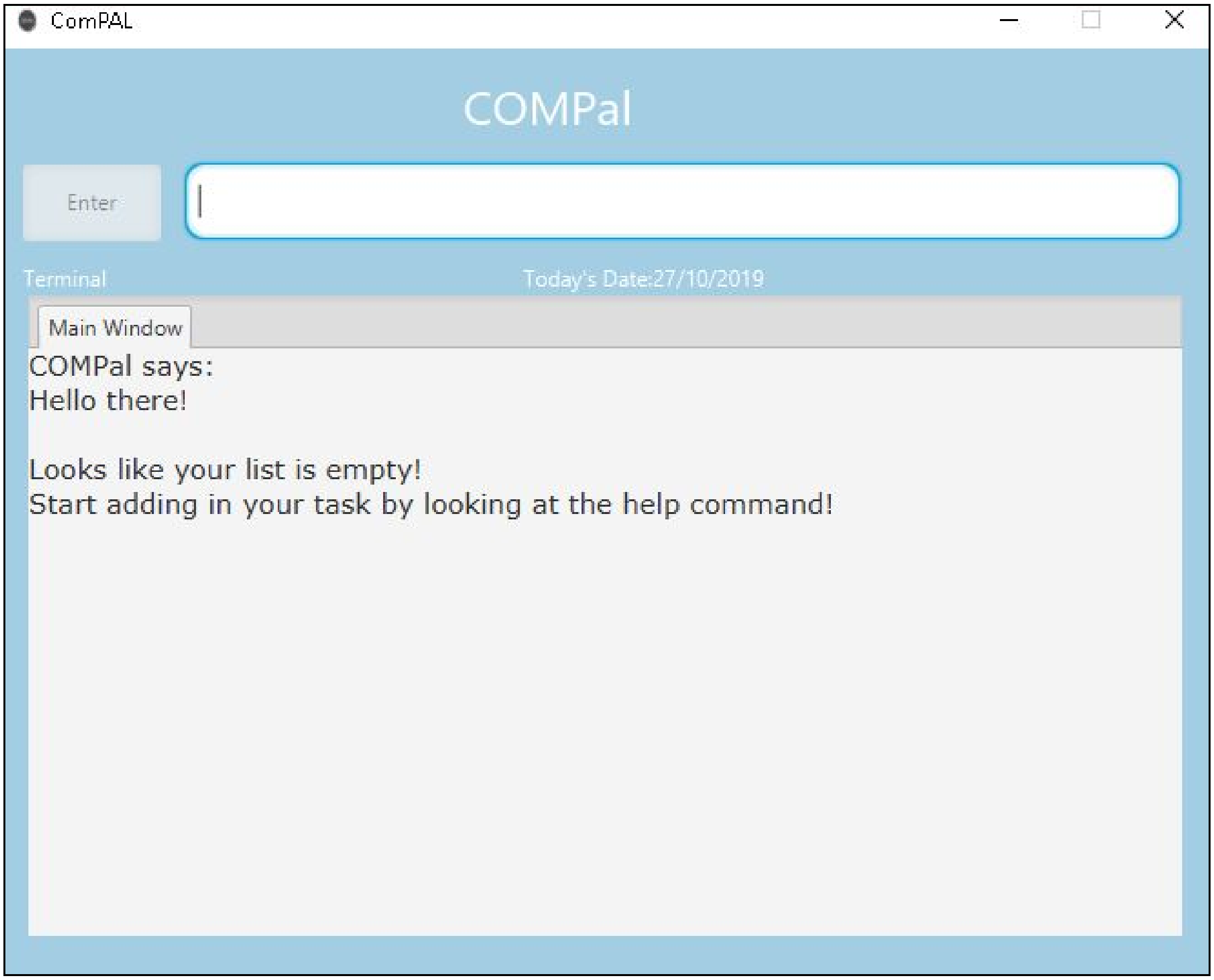
3. Quick Start

1. Ensure you have **Java Version 11** or above installed on your computer.

2. Download the latest version of COMPal.jar here.

3. Copy the COMPal.jar file to the folder that you want to use as the home folder for **COMPal**.

4. Double-click the COMPal.jar file to start the application. The **COMPal** Graphical user interface (GUI) should appear in a few seconds.



```
COMPal says:
Hello there!

Looks like your list is empty!
Start adding in your task by looking at the help command!
```

Figure 2. Graphical user interface screen upon start-up

5. And that's it! **COMPal** just became your newest friend / most awesome assistant. You can now try entering commands in the **command box** and press Enter to execute it.

6. Some example commands you can try:
    - `event DESCRIPTION /date DATE /start START_TIME /end END_TIME [/priority PRIORITY] [/final-date FINAL_DATE]` : **adds** an **event** that occurs at a specific time on a specific date
    - `deadline DESCRIPTION /date DATE /end END_TIME [/priority PRIORITY] [/final-date FINAL_DATE]` : ***adds*** a **task** with a **deadline**.
    - `find KEYWORD`: ***find*** a certain **task** related to a ***keyword***.
    - `list`: ***displays the entire list of*** **tasks** stored in **COMPal**'s massive memory. However, if you have just downloaded **COMPal**, it's likely that **COMPal** will not display any **tasks**.
    - `bye`: ***bid goodbye*** to **COMPal** and exit the program. See you soon!!

⚠	Note that `TASK_DESCRIPTION` is not meant to have the character '_' (underscore) in it.

This is the end of the **Quick Start** guide. Please refer to **Features** for more details on more commands. Enjoy planning your life with **COMPal**!

4. Features

This is the part that you have all been waiting for **- COMPal**'s features! This section contains all the information that you need to know to let **COMPal** manage your life better.

Command Format

- Words in `UPPER_CASE` are the parameters to be supplied by the user e.g. `help COMMAND_NAME`, `COMMAND_NAME` is a parameter which can be used as `help deadline`.
- Items in ***square brackets are optional*** e.g. `view day [/date DATE] [/type TASK_TYPE]` can be used as `view day /date 29/10/2019` or as `view day`.
- Parameters can be in ***any order*** e.g. if the command specifies `/id TASK_ID /status Y/N`, `/status Y/N /id TASK_ID` is also acceptable.
- Items in angle brackets are ***optional***, ***but at least one item*** must be ***included*** e.g. `</description DESCRIPTION> </date DATE>` can be used as `/description go for a run` or `/description go for a run /date 28/10/2019`

4.1. General Commands

This section covers an extensive guideline on the usage of general commands that you can use on **COMPal**!

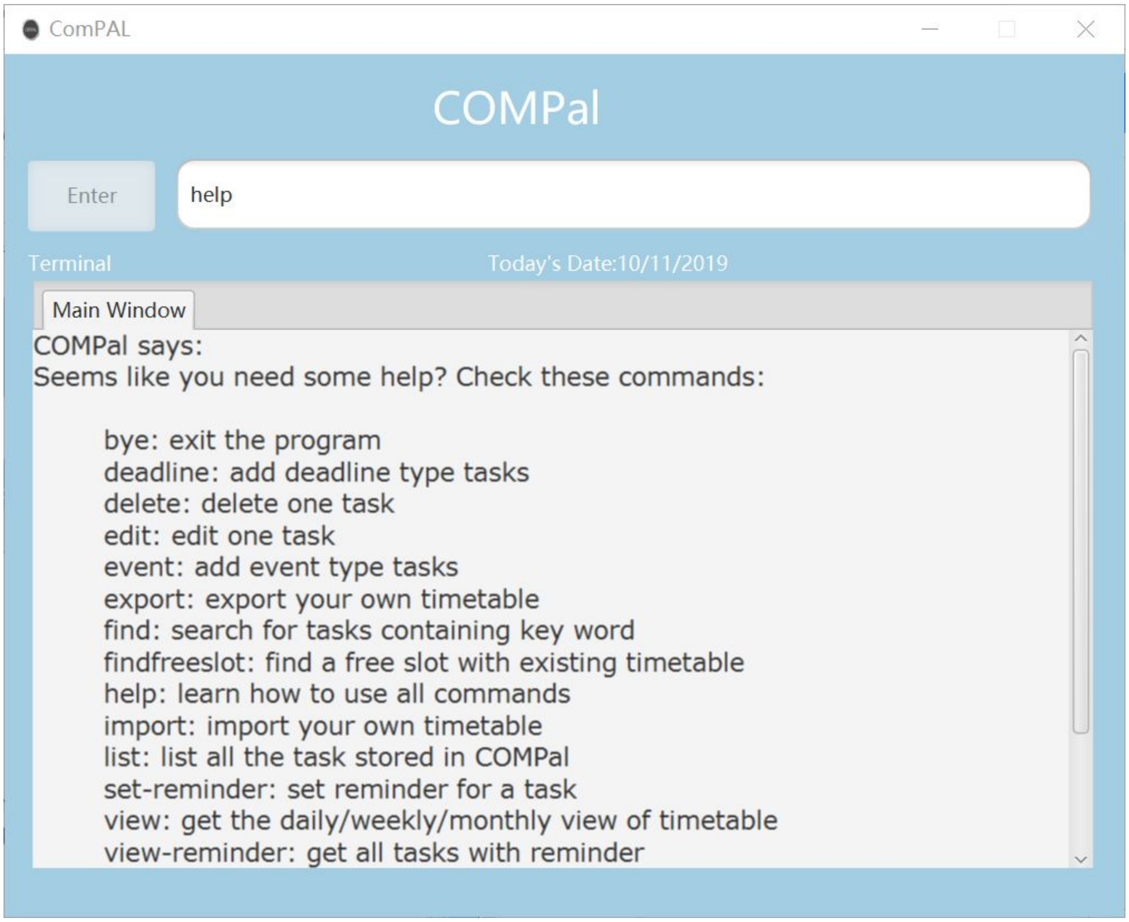
4.1.1. Viewing Help: `help`

Can't remember so many tedious commands?

Format for ***general help***: `help` or `TRASH_COMMAND`

You can see a list of all commands available. There is also a guide to tell you how to use this `help` command.

ℹ	`TRASH_COMMAND` is really "trash command" 😊, it could be any command that is invalid.



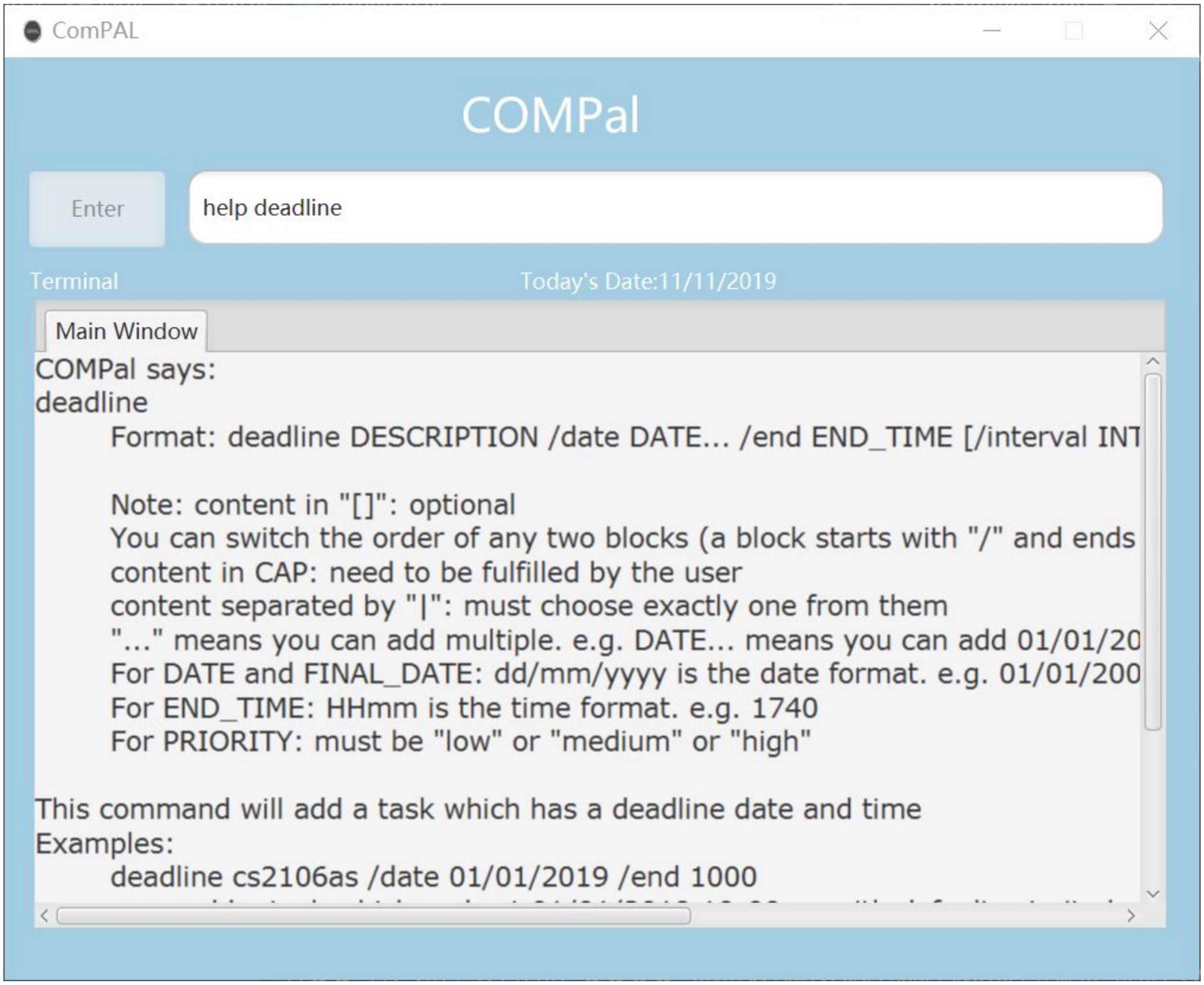
```
COMPal says:
Seems like you need some help? Check these commands:

      bye: exit the program
      deadline: add deadline type tasks
      delete: delete one task
      edit: edit one task
      event: add event type tasks
      export: export your own timetable
      find: search for tasks containing key word
      findfreeslot: find a free slot with existing timetable
      help: learn how to use all commands
      import: import your own timetable
      list: list all the task stored in COMPal
      set-reminder: set reminder for a task
      view: get the daily/weekly/monthly view of timetable
      view-reminder: get all tasks with reminder
```

Figure 3.Output after entering the `help` command.

Format for ***specific help***: `help COMMAND_NAME`

You can use this command to search for specific information about `COMMAND_NAME`.

ℹ	`COMMAND_NAME` is any command name you can see when you do `help` command.



```
COMPal says:
deadline
        Format: deadline DESCRIPTION /date DATE... /end END_TIME [/interval INT

        Note: content in "[]": optional
        You can switch the order of any two blocks (a block starts with "/" and ends
        content in CAP: need to be fulfilled by the user
        content separated by "|": must choose exactly one from them
        "..." means you can add multiple. e.g. DATE... means you can add 01/01/20
        For DATE and FINAL_DATE: dd/mm/yyyy is the date format. e.g. 01/01/200
        For END_TIME: HHmm is the time format. e.g. 1740
        For PRIORITY: must be "low" or "medium" or "high"

This command will add a task which has a deadline date and time
Examples:
        deadline cs2106as /date 01/01/2019 /end 1000
```

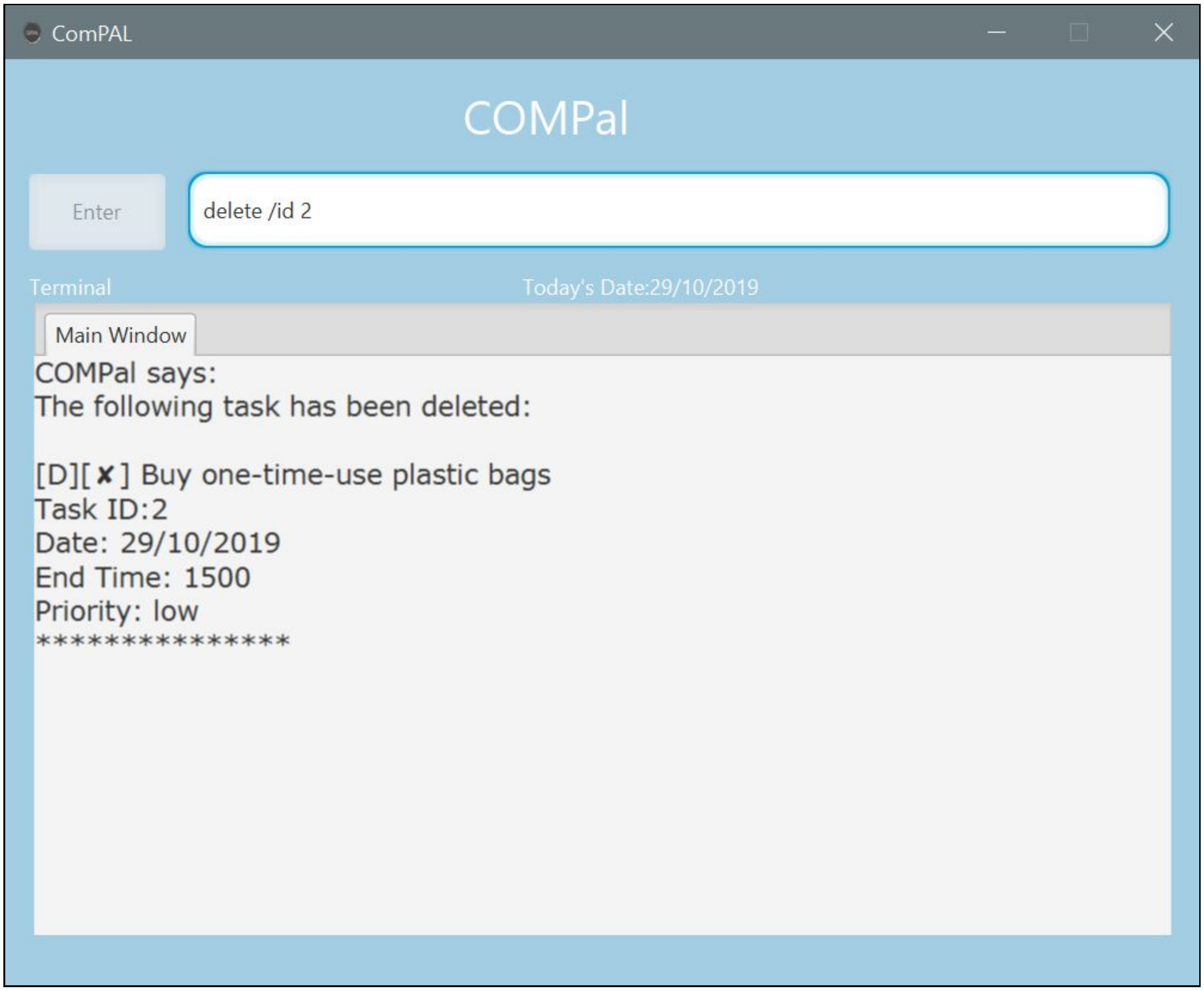
Figure 4. Output after entering `help deadline` command.

4.1.2. Deleting a Task: `delete`

Sometimes, you just don't feel like doing things. Maybe someone pressured you to do something, and after mulling over it for hours, you decided to change your mind. **COMPal** understands that you're human after all - just delete that **task**, and move on with your life.

Format for ***deleting a task***: `delete /id TASK_ID`

ℹ	`TASK_ID` is the **task** ID shown when you do `list` command.



```
delete /id 2
```

```
COMPal says:
The following task has been deleted:

[D][✘] Buy one-time-use plastic bags
Task ID:2
Date: 29/10/2019
End Time: 1500
Priority: low
***************
```

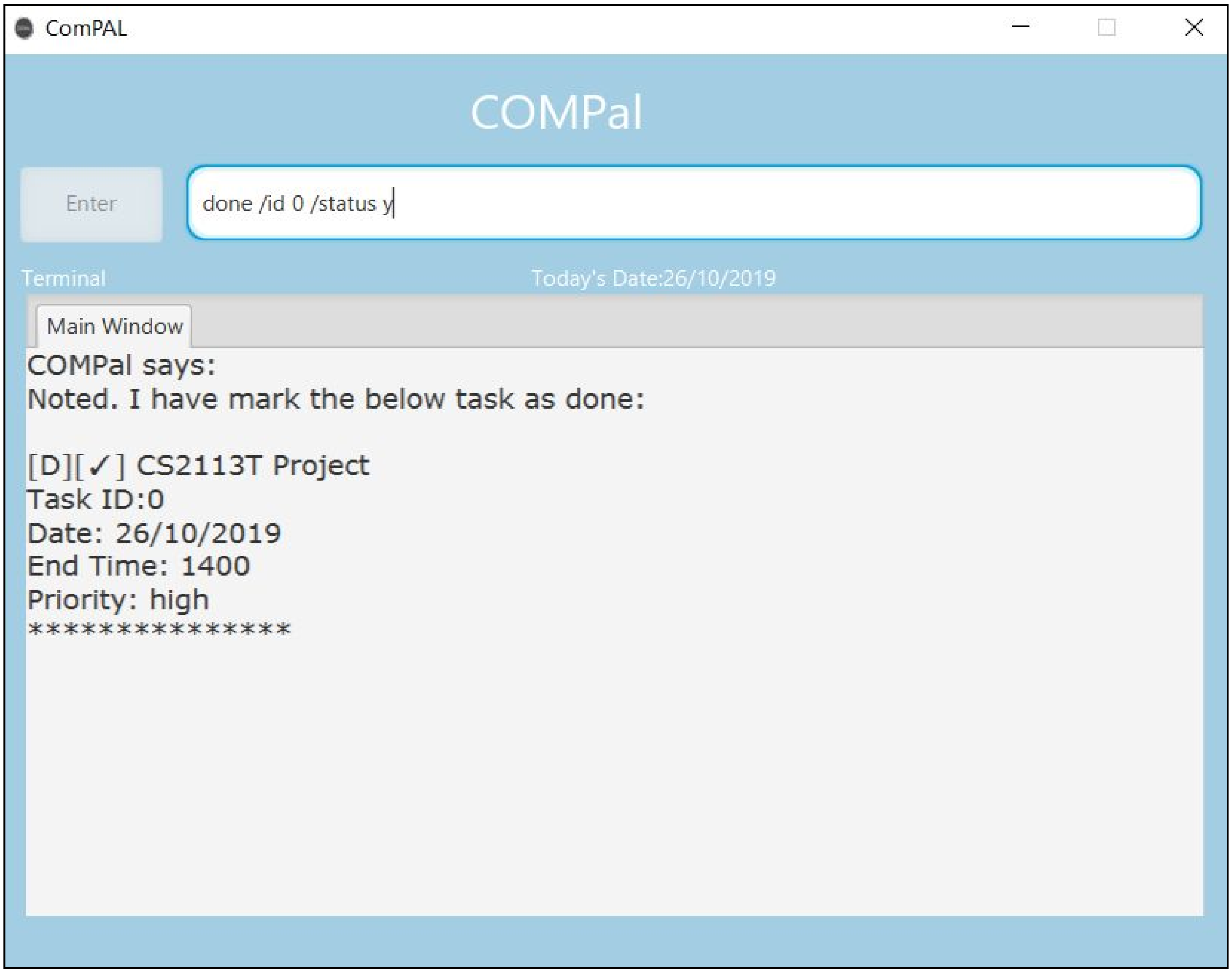
Figure 5. Output after entering the `delete` command.

4.1.3. Changing a Task Status: `done`

After completing a **task**, you may want to mark it as done.

Format for ***marking a task as done***: `done /id TASK_ID /status Y`
Format for ***marking a task as incomplete***: `done /id TASK_ID /status N`

ℹ	`TASK_ID` is the **task** ID shown when you do `list` command.



```
done /id 0 /status y
```

```
COMPal says:
Noted. I have mark the below task as done:

[D][✓] CS2113T Project
Task ID:0
Date: 26/10/2019
End Time: 1400
Priority: high
***************
```

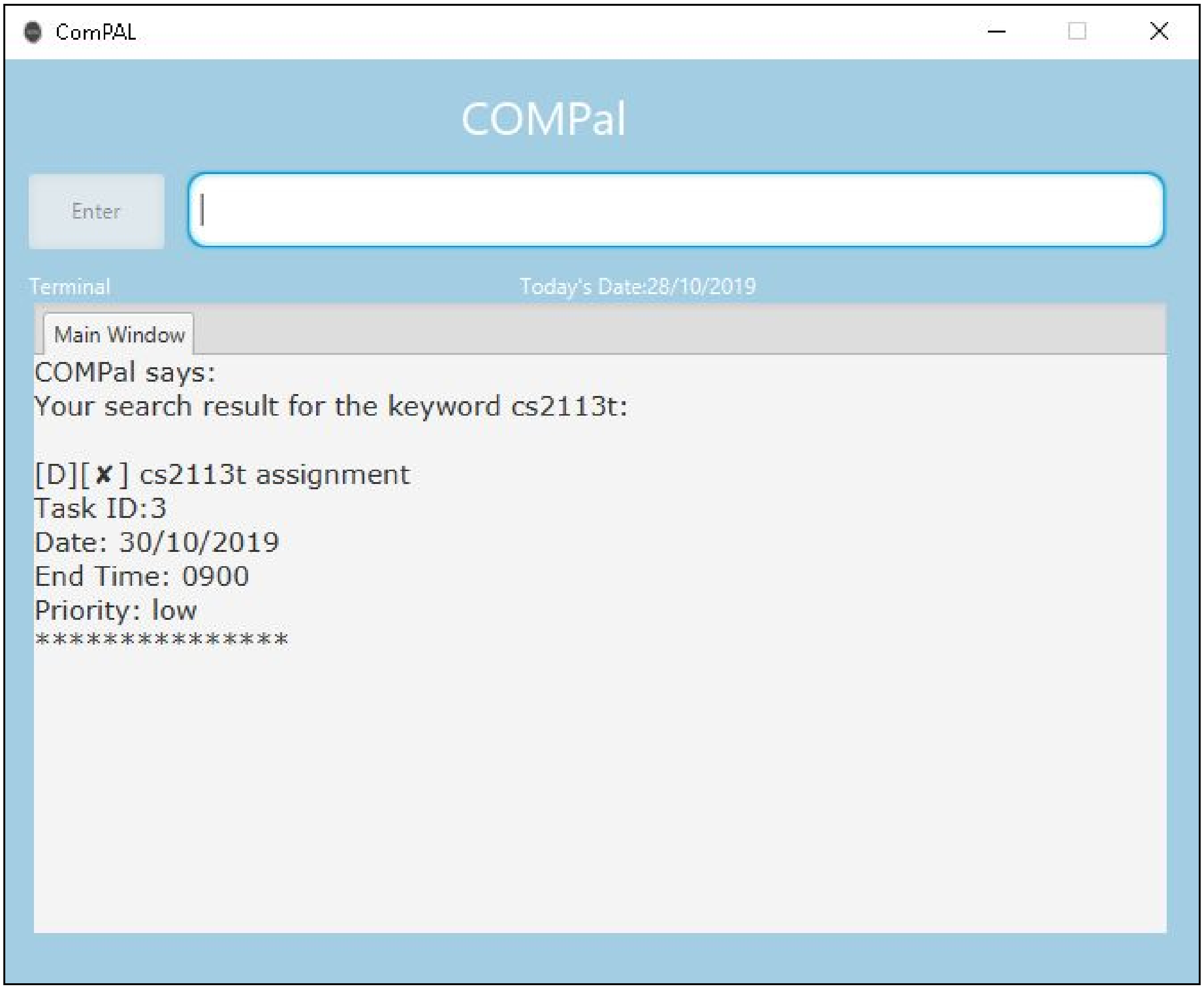
Figure 6. Output after entering the `done` command.

4.1.4. Finding a Task: `find`

Want to search for the **task** by its keyword? Enter `find KEYWORD` in the **command box** will give you all the **tasks** including the keyword.

Format for ***finding a task with `KEYWORD` in its description***: `find KEYWORD`

ℹ	The `find` command without any arguments following it will return and display all **tasks** to you.



```
COMPal says:
Your search result for the keyword cs2113t:

[D][✘] cs2113t assignment
Task ID:3
Date: 30/10/2019
End Time: 0900
Priority: low
***************
```

Figure 7. Output of an example command `find cs2113t`.

4.1.5. Viewing Tasks: `view`

Want to view the **tasks** stored in COMPal in a day, week or monthly view? Enter the `view` command in the **command box**!

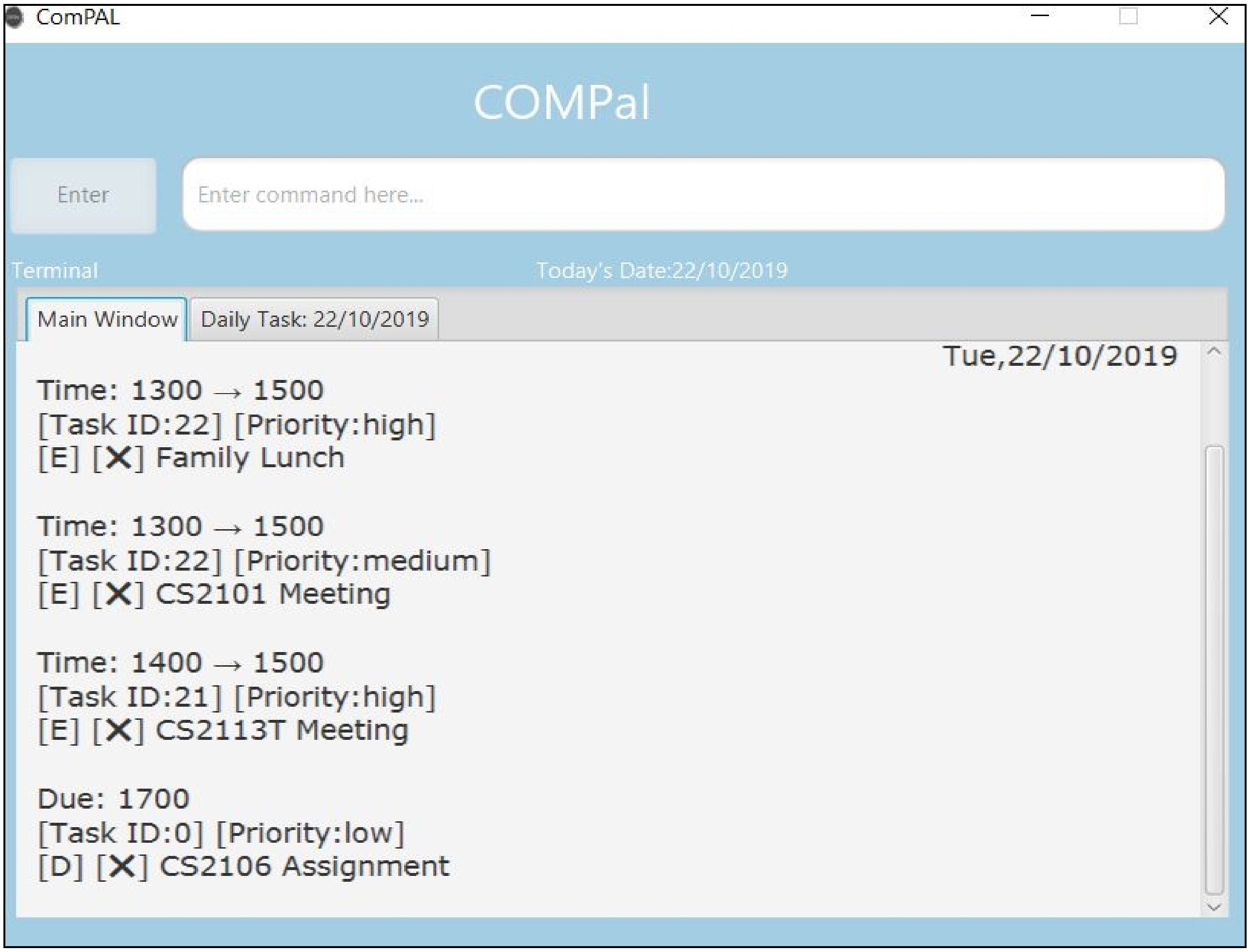
ℹ	DATE is the date you want to search for and should be **DD/MM/YYYY** format. E.g. 02/10/2019.

Format for ***viewing the daily schedule of a date***: `view day [/date DATE]`
Format for ***viewing the weekly schedule of a week date***: `view week [/date DATE]`
Format for ***viewing the monthly schedule of a specified month***: `view month [/date DATE]`
Format for ***viewing specified types within a daily schedule of a date***: `view day [/date DATE] [/type TASK_TYPE]`

💡	You ***may omit*** `DATE` to view the calendar with respect to the current date! E.g. `view week` will allow you to view the current week and `view day` will allow you to view the current day!
ℹ	***Currently***, there are ***only two types*** COMPal is able to process: **events** and **deadlines**.

Examples:

- `view day /date 09/09/2019`
- `view week /date 10/20/2020 /type deadline`
- `view month /type event`



```
                                                     Tue,22/10/2019
Time: 1300 → 1500
[Task ID:22] [Priority:high]
[E] [✕] Family Lunch

Time: 1300 → 1500
[Task ID:22] [Priority:medium]
[E] [✕] CS2101 Meeting

Time: 1400 → 1500
[Task ID:21] [Priority:high]
[E] [✕] CS2113T Meeting

Due: 1700
[Task ID:0] [Priority:low]
[D] [✕] CS2106 Assignment
```

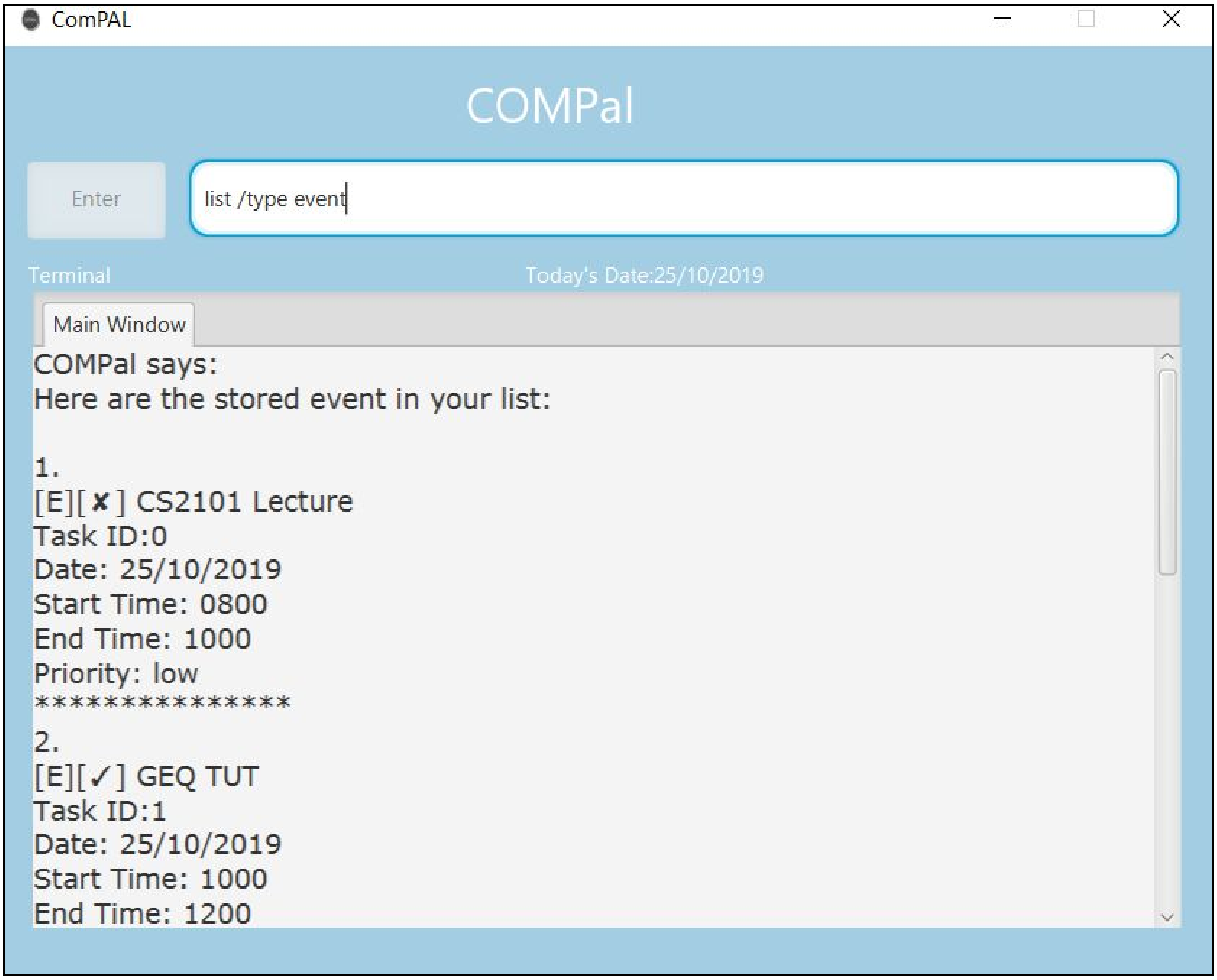
Figure 8. Output of **events** and **deadlines** on the main window in text format, sorted by priority level in a chronological timeline.

4.1.6. Listing Task: `list`

Want to list all ***tasks*** or types of ***tasks*** that are stored in COMPal for past and future ***tasks***? Enter the `list` in the **command box** to see all the **tasks**!

Format for ***listing all tasks stored***: `list`
Format for ***listing specified tasks stored***: `list [/type TASK_TYPE]`
Format for ***listing specified tasks stored with status***: `list [/type TASK_TYPE] [/status done/ongoing/due]`

> 💡 `done` status refers to **tasks** that are marked done by the users.
> `ongoing` status refers to **tasks** that are not marked done and not past its **tasks** starting date.
> `due` status refers to **tasks** that are not marked done and past its start date.



```
list /type event
```

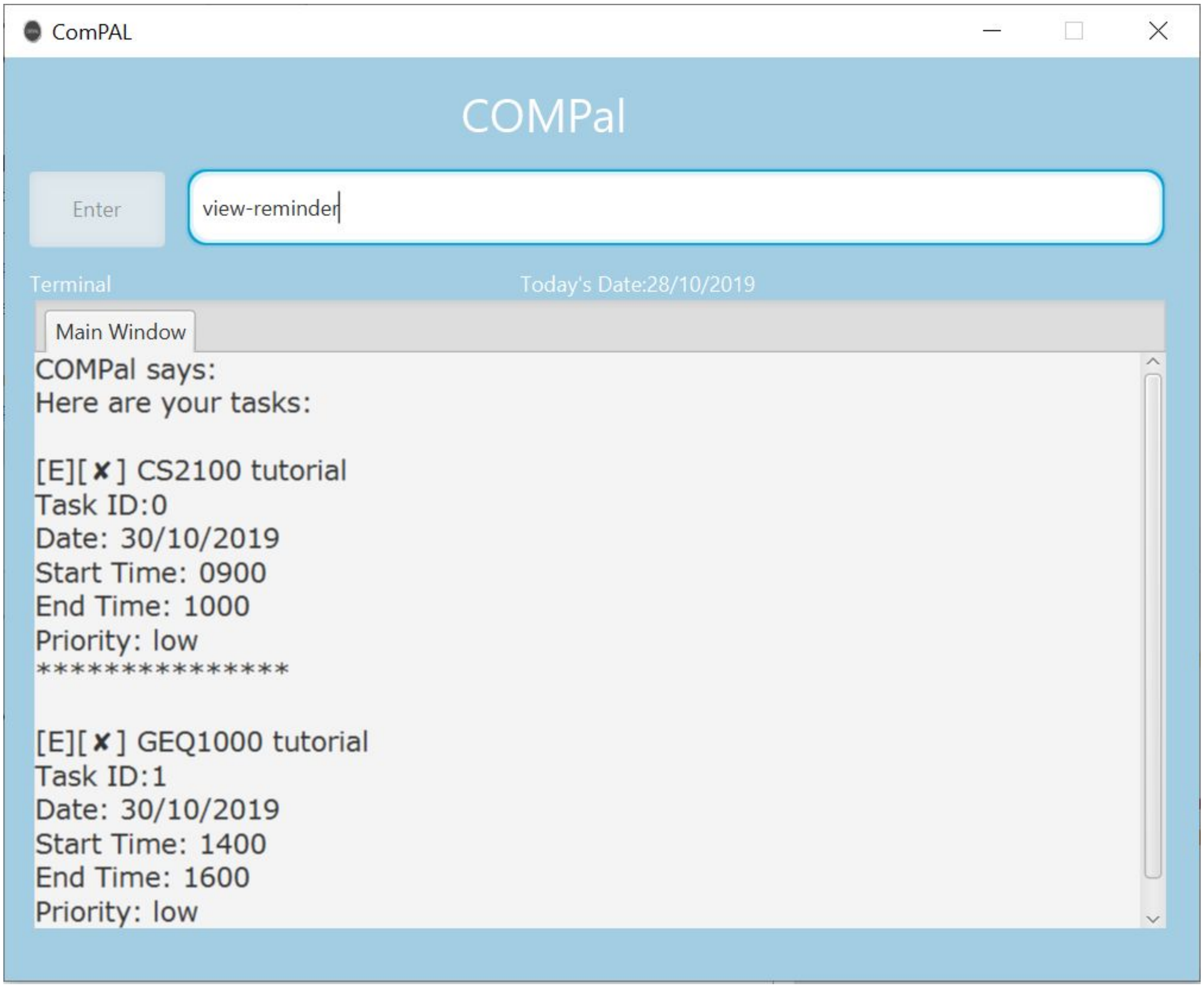
```
COMPal says:
Here are the stored event in your list:

1.
[E][✗] CS2101 Lecture
Task ID:0
Date: 25/10/2019
Start Time: 0800
End Time: 1000
Priority: low
***************
2.
[E][✓] GEQ TUT
Task ID:1
Date: 25/10/2019
Start Time: 1000
End Time: 1200
```

Figure 9. Expected output of `list` command to show all events stored in COMPal.

4.1.7. View Reminder: `view-reminder`

Don't know what is inside this week's reminder list? Enter `view-reminder` in the **command box** to view all reminders this week. Undone **tasks** that are due within the week and overdue **tasks** are pre-set to be included. Additionally, users can manually turn on reminders for important **tasks** they want to keep track of. (Refer to **4.1.8. Set Reminder: `set-reminder`** on how to turn on reminders)

Format for viewing reminders: `view-reminder`



```
COMPal says:
Here are your tasks:

[E][✘] CS2100 tutorial
Task ID:0
Date: 30/10/2019
Start Time: 0900
End Time: 1000
Priority: low
***************

[E][✘] GEQ1000 tutorial
Task ID:1
Date: 30/10/2019
Start Time: 1400
End Time: 1600
Priority: low
```

Figure 10. Expected output of `view-reminder` command.

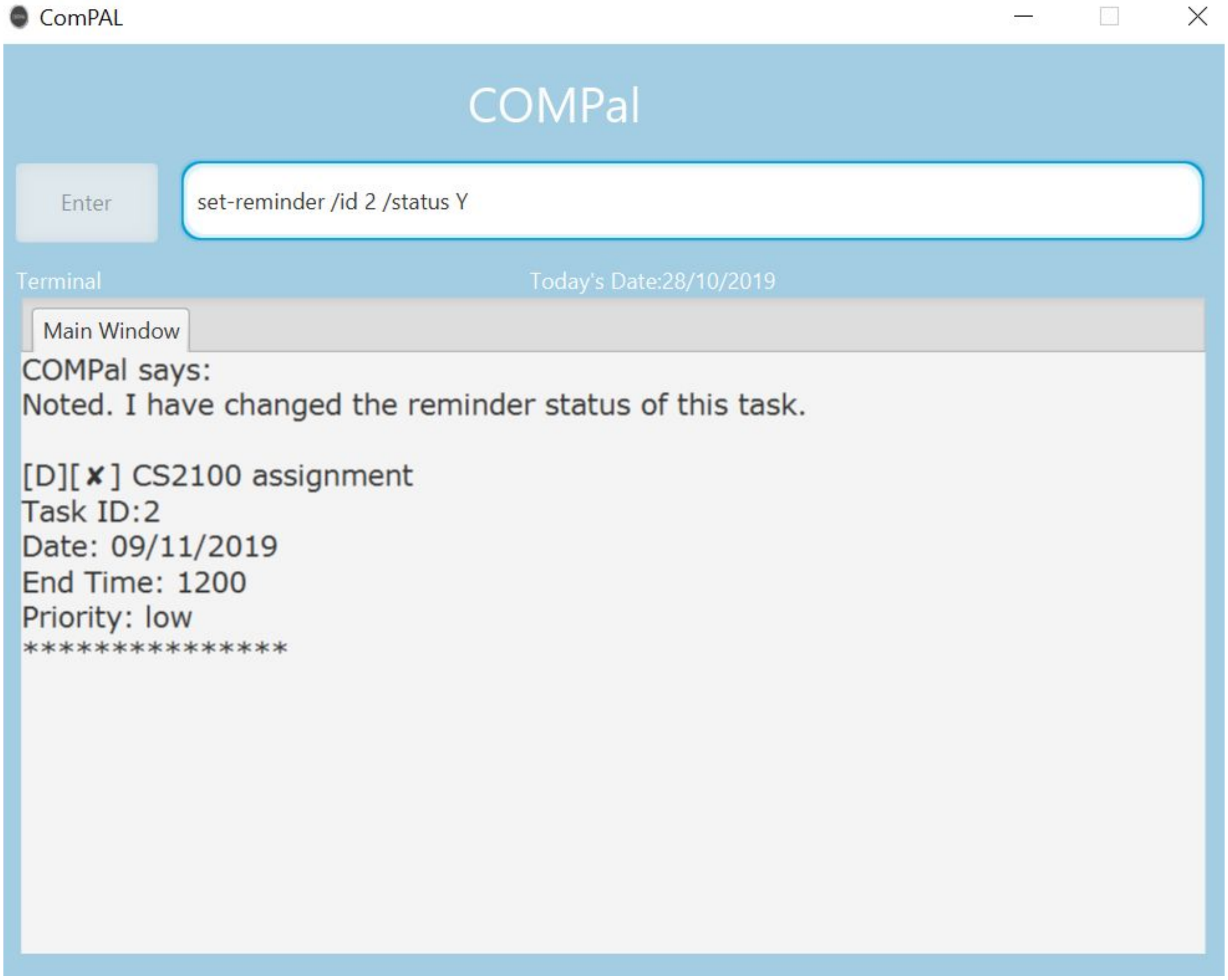
4.1.8. Set Reminder: `set-reminder`

Need a reminder for a **task**? Enter `set-reminder /id TASK_ID /status Y/N` in the **command box** to set a reminder for that **task**.

ℹ	- `TASK_ID` is the **task** ID - `Y/N` is the status of the reminder, to turn the reminder settings on or off respectively

Examples:

- `set-reminder /id 3 /status Y`
- `set-reminder /id 1 /status N`



```
set-reminder /id 2 /status Y
```

```
COMPal says:
Noted. I have changed the reminder status of this task.

[D][✘] CS2100 assignment
Task ID:2
Date: 09/11/2019
End Time: 1200
Priority: low
***************
```

Figure 11. Expected output of `set-reminder /id 2 /status Y` command.

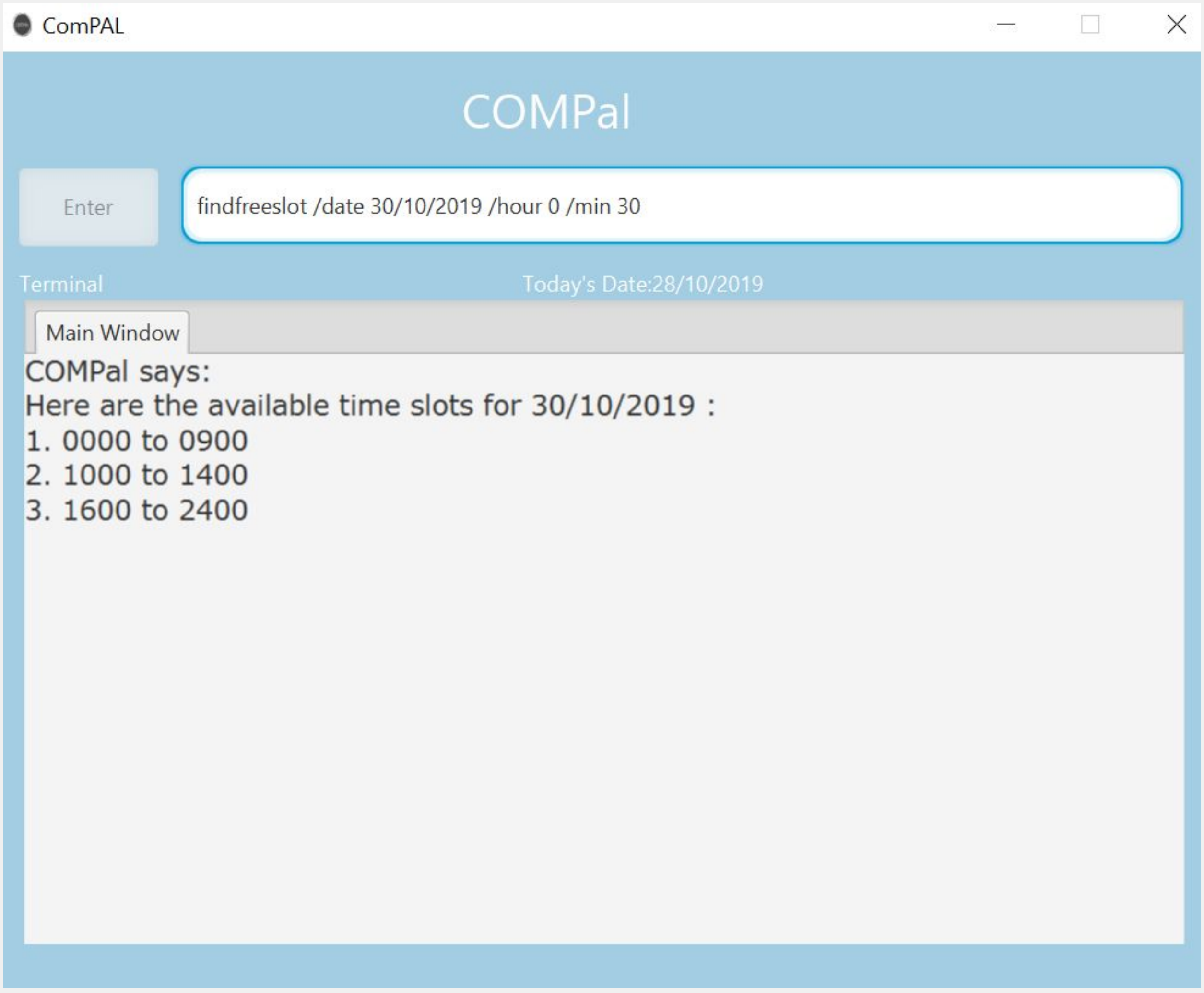
4.1.9. Finding Free Time Slots: `findfreeslot`

Need to find a free time slot to arrange a project meeting? Enter `findfreeslot /date DATE /hour HOUR /min MINUTE` in the **command box** to find a free time slot in the specified date with the specified duration in hours and minutes.

- `DATE` is the date of the free time slot needed
- `HOUR` is the input hour duration of the time slot
- `MIN` is the input minute duration of the time slot

Examples:

- `findfreeslot /date 29/10/2019 /hour 2 /min 0`
- `findfreeslot /date 01/11/2019 /hour 1 /min 30`



```
findfreeslot /date 30/10/2019 /hour 0 /min 30
```

```
COMPal says:
Here are the available time slots for 30/10/2019 :
1. 0000 to 0900
2. 1000 to 1400
3. 1600 to 2400
```

Figure 12. Expected output of `findfreeslot /date 30/10/2019 /hour 0 /min 30` command.

4.1.10. Exporting COMPal Schedule: `export`

Want to export your COMPal schedule to another application? With the export command, you can export it as an iCalender file!

Format: `export /file_name FILE_NAME`

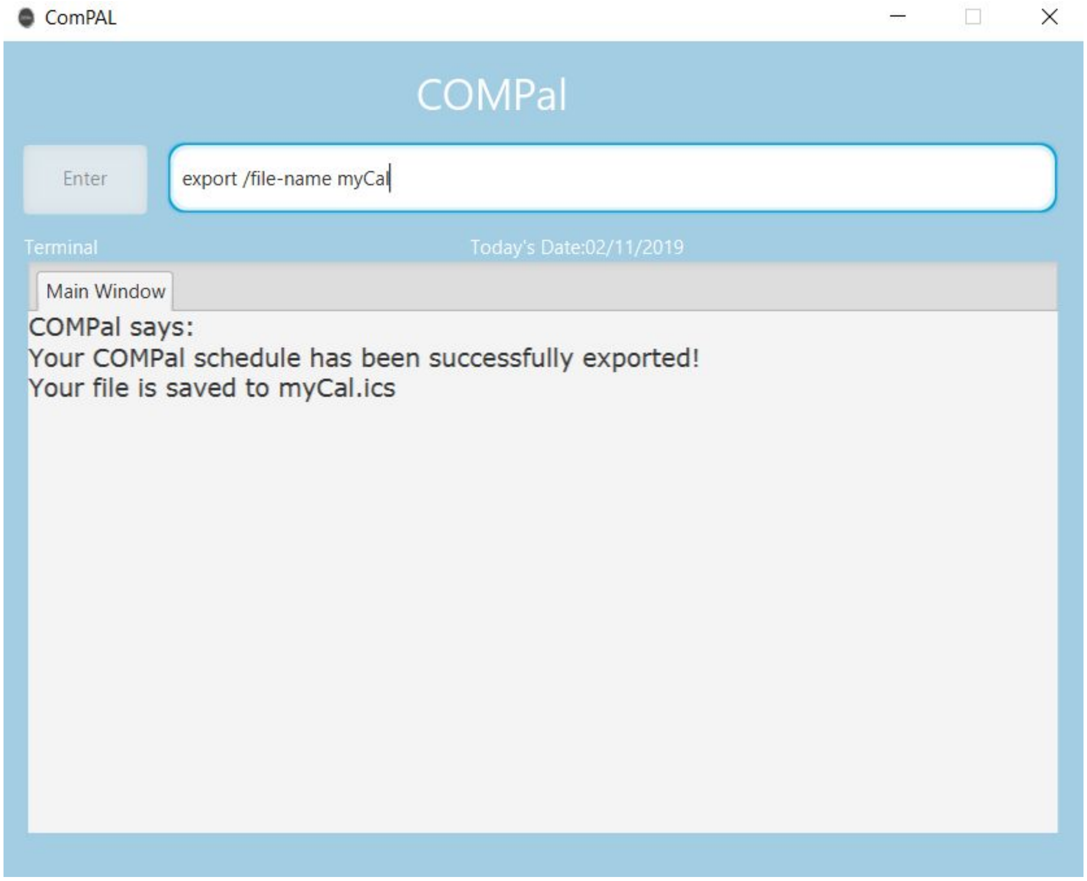
ℹ	Your exported file will be saved in the same folder you run COMPal at!

With this function, you can import your COMPal schedule to another COMPal application or share it with your friends or even import it to google calendar or any calendar that is able to import iCalendar files!

Check this link here to learn how to import to Google Calendar!

Examples:

- `Export /file-name myCal`



```
export /file-name myCal
```

```
COMPal says:
Your COMPal schedule has been successfully exported!
Your file is saved to myCal.ics
```

Figure 13. Expected output of `export /file-name myCal` command.

4.1.11. Importing iCalendar files: `import`

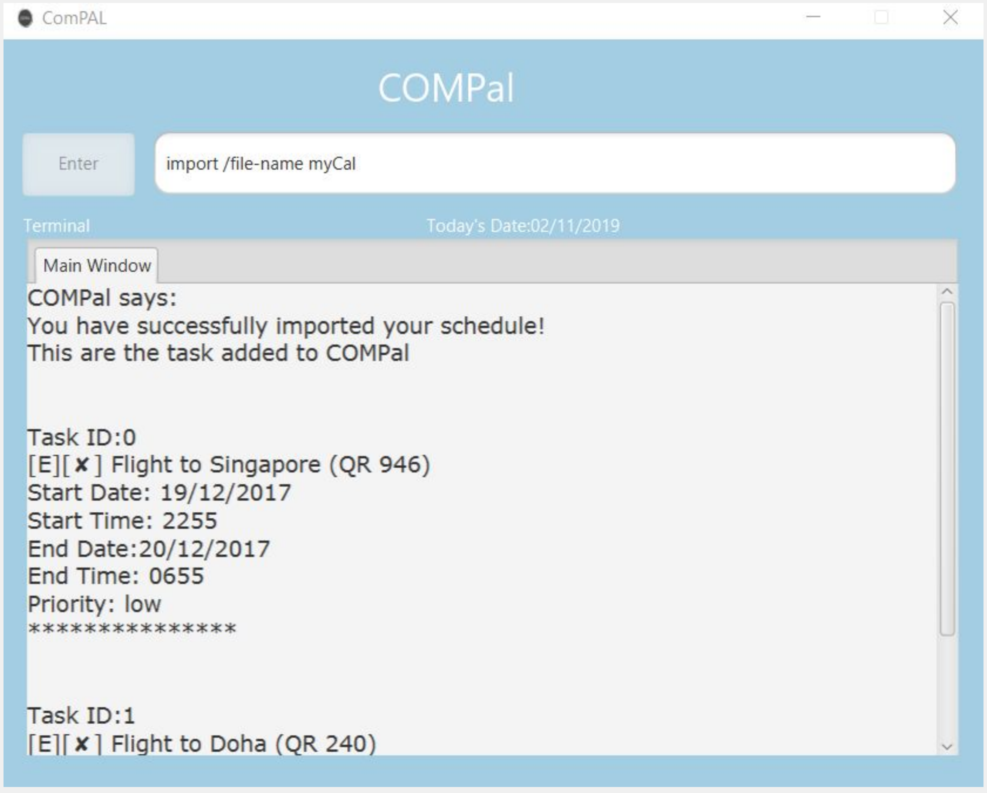
Want to import your COMPal schedule from another computer or maybe other calendar applications you have used into COMPal? Yes, you can import it as long the file is an `iCalendar` file!

Format: `import /file_name FILE_NAME`

Examples:

- `import /file_name myCal`

⚠	File must be in the same directory as **COMPal** launch application for this command to work!



```
import /file-name myCal
```

```
COMPal says:
You have successfully imported your schedule!
This are the task added to COMPal


Task ID:0
[E][✘] Flight to Singapore (QR 946)
Start Date: 19/12/2017
Start Time: 2255
End Date:20/12/2017
End Time: 0655
Priority: low
***************


Task ID:1
[E][✘] Flight to Doha (QR 240)
```

Figure 14. Sample output of `import /file-name myCal` command with iCalendar file generated from google calendar.

⚠	This feature ***works best*** with the `iCalendar` file generated from COMPal. However, for `iCalendar` files that are not generated by COMPal, It will be imported if the event ***has a start and end time with no rules and timezone attached.*** Additionally, added events priority will be set to low for importing `iCalendar` not

	generated from COMPal.

4.1.12. Exiting COMPal: `bye`

Bye-Bye! Enter bye in the **command box** will quit **COMPal**. Have a nice day! 😊

4.2. Detailed Task Management Commands

The Merriam-Webster's dictionary defines a **task** as “a usually assigned piece of work, often to be finished within a certain time”. **COMPal** has a similar definition - if you have something to do, you can track it as a **task** using **COMPal**.

COMPal accepts two main types of **tasks**:

1. A **deadline** is a **task** that has to be done by a **specific time** on a **specific date**.
2. An **event** is a **task** that has to be done during a **fixed duration** on a **specific date**.

Furthermore, **COMPal** is able to manage **tasks** that **recur** i.e. happen at **regular intervals**, such as a weekly project meeting or a weekly Tutorial session.

Since these two **tasks** behave very differently, we have separate commands to manage them:

1. `deadline`: For adding **deadlines**.
2. `event`: For adding **events**.
3. `edit`: For editing either **events** or **deadlines**.

Each **task** type has a system of short and simple **parameter keywords**, which will help **COMPal** process your input just the way you like it.

4.2.1. Deadline Management

Dread them. Run from them. But **deadlines** still arrive.

In this section, you will be introduced to commands and parameters that help you manage those pesky **deadlines** that seem to suck all the joy out of your life.

Below is a list of **parameters** and **keywords** that you can expect to use for the commands in this section.

Table 1: Parameters and **keywords** for **deadlines**.

Keyword	Parameter	Usage
-	`DESCRIPTION`. Underscores '_' are not allowed in the description.	You can ***describe*** your **deadline** in any detail. No keyword is required to be typed before your input - just describe your **deadline**!!
`/date`	`DATE`, in the format `DD/MM/YYYY`	You can enter the ***date*** that your **deadline** has to be completed ***by***
`/end`	`END_TIME`, in the format `HHmm`	You can enter the ***time*** that the **deadline** has to be completed ***by***
`/priority`	`PRIORITY` (`low`, `medium`, `high`)	You can assign a ***priority*** to multiple/single **deadline(s)**.
`/final-date`	`FINAL_DATE`, in the format `DD/MM/YYYY`	You can use this to add ***multiple*** **deadlines** that occur at ***regular intervals***. `FINAL_DATE` will be taken as the latest possible date for your ***final*** **deadline**.
`/interval`	`INTERVAL`, positive number greater than `zero`	You can use this to add ***multiple*** **deadlines** that occur at ***regular intervals***. `Interval` will be taken as the interval between each recurring **deadline**, in terms of the number of days.

ℹ	`/final-date`, `/interval` and `/priority` are ***optional keywords***. You can use them for more control over your **tasks** but can leave them out if you want to. **COMPal** will then revert to ***default values***, which will be elaborated in specific commands below.

⚠	Any dates that you enter ***has*** to be in the format `DD/MM/YYYY`, or **COMPal** will not understand your dates!
⚠	Any time that you enter ***has*** to be in the format of `HHmm`, or **COMPal** will be confused!

4.2.1.1. Adding Deadlines

You can use the `deadline` command to get **COMPal** to add impending **deadlines** to its impressive memory, and keep track of them for you. You'll never miss a **deadline** this way!

Basic Command Format:

```
deadline DESCRIPTION /end END_TIME /date DATE
```

Example:

- `deadline Submit CS2113T User Guide for Review /date 02/10/2019 /end 2359`
  Adds a **deadline** with `Submit CS2113T User Guide for Review` as `DESCRIPTION`, `2359` as the `END_TIME` and `02/10/2019` as the `DATE`.

However, the above is merely the ***basic format***. As students, we know that school gives us heaps of assignments and projects to do on a regular basis. As that mountain of work piles up steadily, we invariably have to pick some to prioritise above others. To handle these concerns, **COMPal** lets you assign ***priorities*** and also create ***recurring*** **deadlines**.

4.2.1.1.1. Assigning Priorities to Deadline

The optional `/priority` keyword lets you assign a **deadline** with a `PRIORITY`. If you have a **deadline** that you ***absolutely have*** to complete soon, you can enter a `PRIORITY` of `high`. If your **deadline** isn't that urgent, you can use the value of `medium`, or if it isn't something worth worrying about, you can assign it as `low`.

Alternatively, if you do not use the `/priority` keyword, **COMPal** will set the `PRIORITY` of your **deadline** as `low` by ***default***.

Command Format (Priority):

```
deadline DESCRIPTION /end END_TIME /date DATE /priority PRIORITY
```

Examples:

- `deadline Submit CS2106 Assignment /end 2359 /date 02/10/2019 /priority high`

  Adds a **deadline** with `Submit CS2106 Assignment` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`, and `high` as `PRIORITY`**.**

- `deadline Finish CS2105 Tutorial /end 2300 /date 02/10/2019 /priority medium`

  Adds a **deadline** with `Finish CS2105 Tutorial` as `DESCRIPTION`, `2300` as the `END_TIME`, `02/10/2019` as the `DATE`, and `medium` as `PRIORITY`**.**

- `deadline Submit GEQ1000 Reflection /end 2359 /date 02/10/2019 /priority low`

  Adds a **deadline** with `Submit GEQ1000 Reflection` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`, and `low` as `PRIORITY`**.**

- `deadline Submit GEQ1000 Reflection /end 2359 /date 02/10/2019`

  This adds a **deadline** with `Submit GEQ1000 Reflection` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`.

  However, omitting the `/priority` keyword prompts **COMPal** to automatically assign `low` as `PRIORITY`, meaning that the result of this command is ***identical*** to the previous command. This example illustrates the ***optional*** nature of the `/priority` keyword.

4.2.1.1.2. Adding Recurring Deadlines

The optional `/final-date` and `/interval` keywords work in tandem to let you add multiple ***recurring*** **deadlines**.

`/final-date` allows you to recur **deadlines** until a specified `FINAL_DATE`, which is the ***final possible date*** for your final **deadline**. If you do not use the `/final-date` keyword, **COMPal** assumes that you are not adding ***recurring*** **deadlines**.

`/interval` allows you to specify the `INTERVAL` between each recurring **deadline**, in terms of the number of days. If you do not use the `/interval` keyword, **COMPal** will set your `INTERVAL` to `7` by ***default*** i.e. every **deadline** occurs exactly ***seven days / one week*** after the previous, as long as it is not after the `FINAL_DATE` specified using the `/final-date` keyword.

Of course, in life, there's also a chance of ***regular, but not identical*** intervals between **deadlines**. **COMPal** introduces a neat little hack: if you add multiple `DATE`s after your `/date` keyword, **COMPal** will add recurring **deadlines** for each `DATE` using the `INTERVAL` (specified or default), as long as it is not after the `FINAL_DATE` specified using the `/final-date` keyword.

Command Format (Recurring):

```
deadline DESCRIPTION /end END_TIME /date DATE /final-date FINAL_DATE
/interval INTERVAL
```

ℹ	You can use the `/priority` keyword here as well. Every ***recurring*** **deadline** will have the same `PRIORITY`.

Examples:

- `deadline Submit CS2106 Assignment /end 2359 /date 02/10/2019`

  Adds a **deadline** with `Submit CS2106 Assignment` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`**.**

  This is the command with a ***basic format***. The following examples will show how adding `/final-date` and `/interval` changes the result.

- `deadline Submit CS2106 Assignment /end 2359 /date 02/10/2019 /final-date 15/12/2019`

  Adds a **deadline** with `Submit CS2106 Assignment` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`, and `15/12/2019` as the `FINAL_DATE`. Omitting the `/interval` keyword means that your `INTERVAL` is set to `7` by ***default***. This means that:

    1. 2359, 02/10/2019, Wednesday is the ***first*** **deadline** where you submit your very first CS2106 assignment.
    2. **COMPal** then creates a **deadline** reminding you to submit your CS2106 assignments for ***every*** Wednesday after that, until the last Wednesday before 15/12/2019 (which is 11/12/2019).

- `deadline Submit CS2106 Assignment /end 2359 /date 02/10/2019 /final-date 15/12/2019 /interval 14`

  Adds a **deadline** with `Submit CS2106 Assignment` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`, `15/12/2019` as the `FINAL_DATE`, and `14` as the `INTERVAL`. This means that:

    1. 2359, 02/10/2019, Wednesday is the ***first*** **deadline** where you submit your very first CS2106 assignment.
    2. **COMPal** then creates a **deadline** reminding you to submit your CS2106 assignments for ***every alternate*** Wednesday (***two weeks*** after the previous Wednesday), until the last ***alternate*** Wednesday that is before 15/12/2019 (which is 27/11/2019).

- `deadline Submit CS2106 Assignment /end 2359 /date 02/10/2019 04/10/2019 /final-date 15/12/2019 /interval 14`

  Adds a **deadline** with `Submit CS2106 Assignment` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` and `04/10/2019` as the `DATE`, `15/12/2019` as the `FINAL_DATE`, and `14` as the `INTERVAL`. This means that:

    1. 2359, 02/10/2019, Wednesday and 2359, 04/10/2019, Friday are the ***first*** **deadline** ***iterations*** where you submit your first few CS2106 assignments.
    2. **COMPal** then creates a **deadline** reminding you to submit your CS2106 assignments for ***every alternate*** Wednesday (***two weeks*** after the previous Wednesday) and ***every alternate*** Friday (***two weeks*** after the previous Friday), until the last ***alternate*** Friday that is before 15/12/2019 (which is 29/11/2019).

- `deadline Submit CS2106 Assignment /end 2359 /date 02/10/2019 /final-date 15/12/2019 /priority high`

  Adds a **deadline** with `Submit CS2106 Assignment` as `DESCRIPTION`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`, `15/12/2019` as the `FINAL_DATE`, and `high` as the `PRIORITY`. This means that:

    1. 2359, 02/10/2019, Wednesday is the ***first*** **deadline** where you submit your very first CS2106 assignment.

2. **COMPal** then creates a **deadline** reminding you to submit your CS2106 assignments for ***every*** Wednesday after that, until the last Wednesday before 15/12/2019 (which is 11/12/2019). For ***every*** **deadline**, the `PRIORITY` is `high`.

💡	Adding multiple `DATE`s without using `/final-date` will still add multiple **deadlines** with the specified `DATE`s. This may be helpful if your **deadlines** do not recur at regular intervals, and somehow you have outrageous photographic memory and remember the exact dates of all your **deadlines**.

⚠	If you add multiple `DATE`s, ensure that your `FINAL-DATE` is after your latest `DATE`, otherwise, any **deadline** with a `DATE` after your `FINAL-DATE` will not be added.

Full Command Format (With all keywords and parameters):

```
deadline DESCRIPTION /end END_TIME /date DATE [/final-date FINAL_DATE]
[/interval INTERVAL] [/priority PRIORITY]
```

💡	Did you know that you do not have to follow the ***exact order*** in the format above? **COMPal** is able to detect the keywords in any order.

⚠	Despite the above tip, do note that your `DESCRIPTION` ***MUST*** be after your very first command word, i.e. `deadline DESCRIPTION`.
⚠	Take care not to enter any `/` into the command box, as **COMPal** may mistake it as an ***invalid parameter***!!!!

4.2.1.2. Editing Deadlines: `edit`

Need to change information about a **deadline**? Enter `edit /id TASK_ID` where TASK_ID is the **task's** id number.

Command Format:

- Editing `DESCRIPTION`: `edit /id TASK_ID /description NEW_DESCRIPTION`
- Editing `DATE`: `edit /id TASK_ID /date NEW_DATE`
- Editing `PRIORITY`: `edit /id TASK_ID /priority NEW_PRIORITY`
- Editing `END_TIME`: `edit /id TASK_ID /end NEW_END_TIME`

You can edit more than one keyword with a single command.

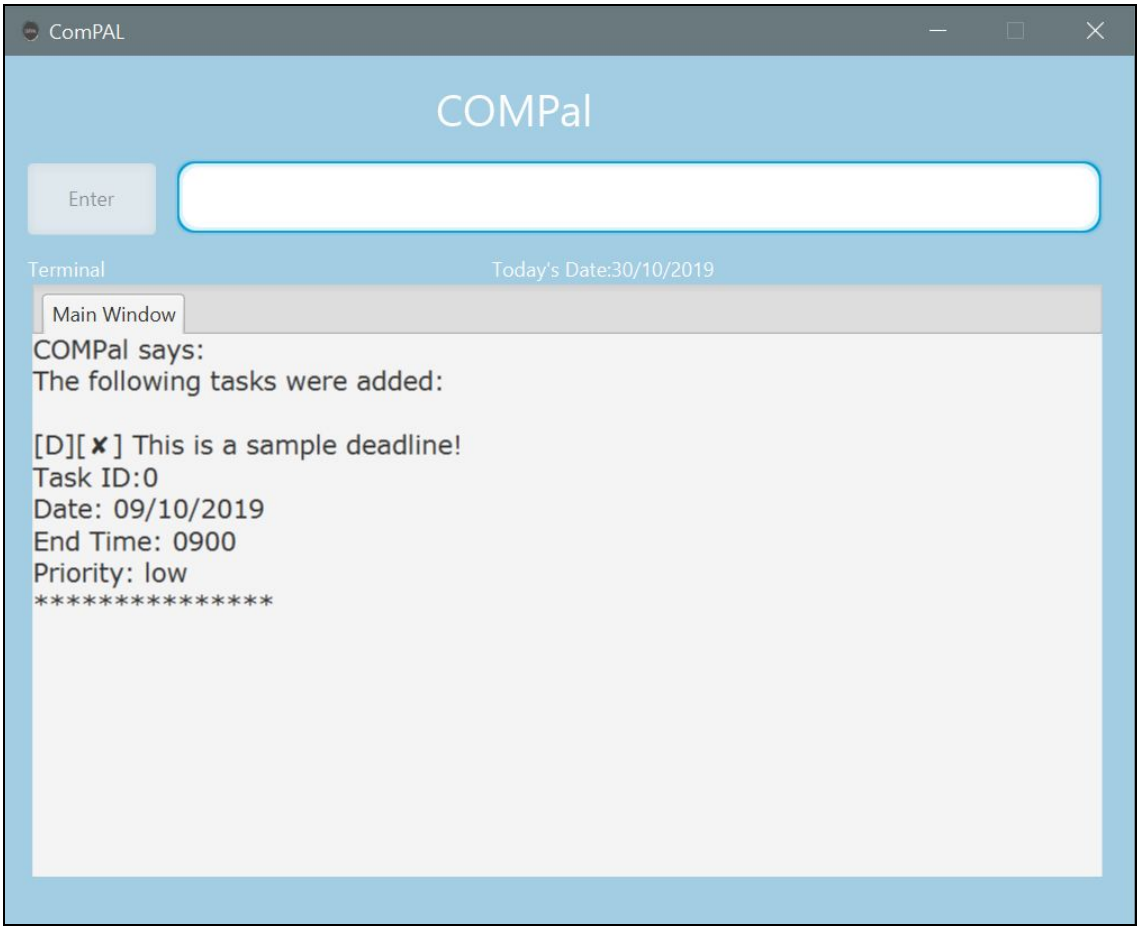
Full Command Format (including all possible keywords):

```
edit /id TASK_ID </description NEW_DESCRIPTION> </end NEW_END_TIME> </date
NEW_DATE> </priority NEW_PRIORITY>
```

⚠	You ***CANNOT*** change `FINAL-DATE` or `INTERVAL`. These parameters are only for ***adding multiple*** **deadlines**, they are not properties of each **deadline**. You can, however, delete any **deadlines** that you think are not necessary anymore or edit each **deadline**.

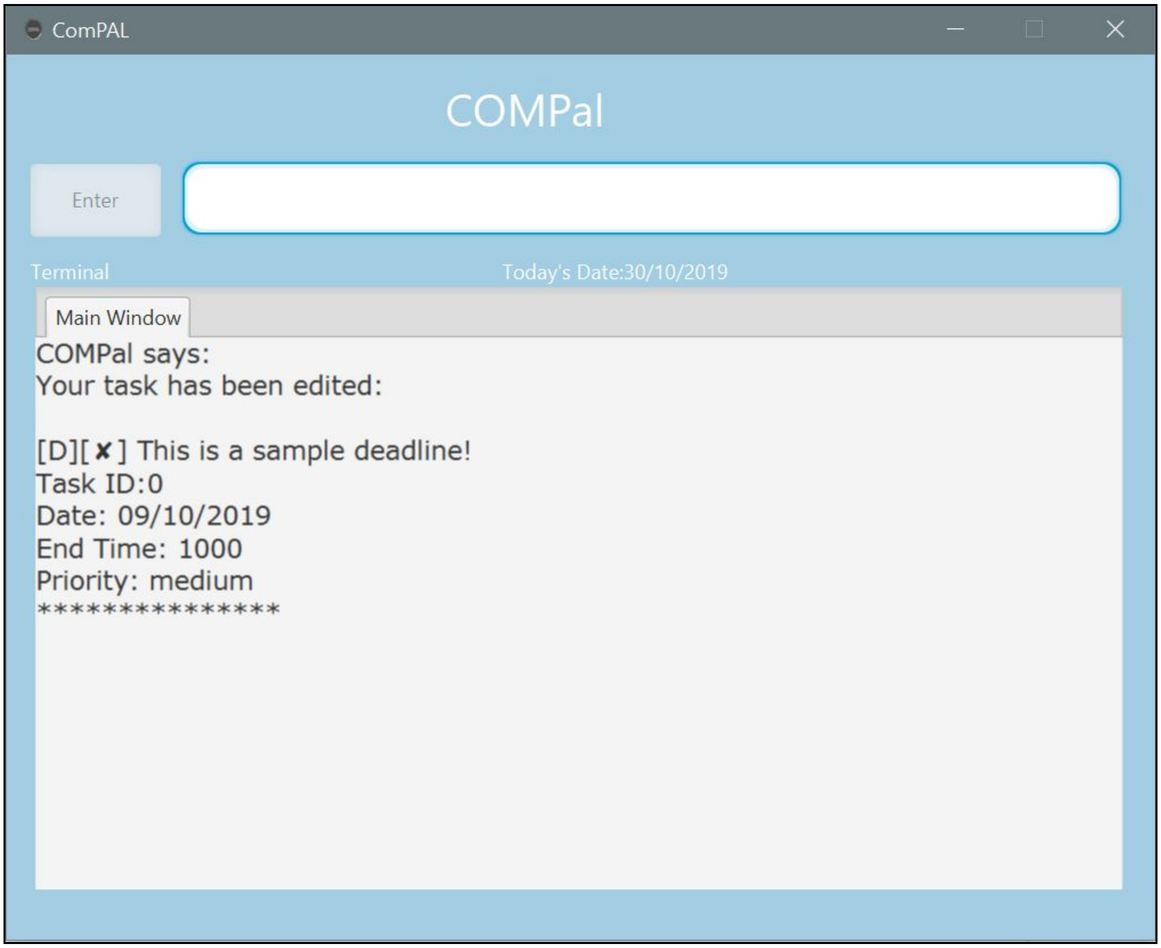
For example, if you wish to increase the `PRIORITY` of a **deadline** with an `ID` of `0` from `low` to `medium` and push back its `END_TIME` from `0900` to `1000`, do:

```
edit /id 0 /end 1000 /priority medium
```



```
COMPal says:
The following tasks were added:

[D][✘] This is a sample deadline!
Task ID:0
Date: 09/10/2019
End Time: 0900
Priority: low
***************
```

Figure 15. The original **deadline** with `TASK_ID` of `0` before the `edit` command executes



```
COMPal says:
Your task has been edited:

[D][✘] This is a sample deadline!
Task ID:0
Date: 09/10/2019
End Time: 1000
Priority: medium
***************
```

Figure 16. The **deadline** with `TASK_ID` of `0` after the `edit` command executes

Note that `PRIORITY` is changed from `low` as shown in Figure 14 to `medium` in Figure 15, and `START_TIME` has been changed from `0900` to `1000,` as intended.

💡	**COMPal** stores your **deadlines** in a `tasks.txt` file, in clear text form. If you are curious by nature and love looking at raw data, you can edit your **deadlines** directly in the `tasks.txt` file.

⚠	While you can edit **deadlines** directly and show your friends that you “hacked” **COMPal**, you cannot add **deadlines** directly in the `tasks.txt` file. You can only use the `deadline` command to add **deadlines**.
⚠	If you read `tasks.txt` closely, you may notice that the various parameters of each **deadline** (such as `DESCRIPTION`, `PRIORITY`, etc.) are separated by underscores (_). ***Do not remove*** these underscores and ensure that each parameter corresponds to the correct type (e.g. `PRIORITY` must always be `low`, `medium`, or `low`, etc.), otherwise **COMPal** will display undefined behaviour.

4.2.2. Event Management

Your best friend's birthday party. Your sibling's graduation. Your cousin's wedding.

Your 8 am Lecture. Your 8-hour code sprint for your software development project. Your compulsory torture session in the frigid exam hall.

Your student life is an endless merry-go-round of things to do - some joyful, some agonising. To better manage your time, you can set a preferred duration for each **task** you have to accomplish, and let **COMPal** track them as **events**. In this section, you will be introduced to commands and parameters that help you manage these **events**.

Below is a list of **parameters** and **keywords** that you can expect to use for the commands in this section.

Table 2: Parameters and **keywords** for **events**.

Keyword	Parameter	Usage
-	`DESCRIPTION`. All characters except underscores can be used.	You can **describe** your **task** in any detail. No keyword is required to be typed before your input - just describe your **event**!!
`/date`	`DATE`, in the format `DD/MM/YYYY`	You can enter the ***date*** that your **event** is happening ***on***
`/start`	`START_TIME`, in the format `HHmm`	You can enter the ***time*** that the **event** is starting ***at***
`/end`	`END_TIME`, in the format `HHmm`	You can enter the ***time*** that the **event** is ending ***at***
`/priority`	`PRIORITY` (`low`, `medium`, `high`)	You can assign a ***priority*** to multiple/single **event(s)**.
`/final-date`	`FINAL_DATE`, in the format `DD/MM/YYYY`	You can use this to add ***multiple*** **events** that occur at ***regular intervals***. `FINAL_DATE` will be taken as the latest possible date for your ***final*** **event**.
`/interval`	`INTERVAL`, positive number greater than `zero`	You can use this to add ***multiple*** **events** that occur at ***regular intervals***. `Interval` will be taken as the interval between each recurring **event**, in terms of the number of days.

ℹ	`/final-date`, `/interval` and `/priority` are ***optional keywords***. You can use them for more control over your **tasks** but can leave them out if you want to. **COMPal** will then revert to ***default values***, which will be specified below.

⚠	Any dates that you enter ***has*** to be in the format `DD/MM/YYYY`, or **COMPal** will not understand your dates!
⚠	Any time that you enter ***has*** to be in the format of `HHmm`, or **COMPal** will be confused!

4.2.2.1. Adding Events

You can use the `event` command to get **COMPal** to add impending **events** to its impressive memory, and keep track of them for you. You'll never miss an **event** this way!

Basic Command Format:

```
event DESCRIPTION /start START_TIME /end END_TIME /date DATE
```

Example:

- `event Dance Practice /start 1800 /end 2000 /date 02/10/2019`

  Adds an **event** with `Dance Practice` as `DESCRIPTION`, `1800` as the `START_TIME`, `2359` as the `END_TIME` and `02/10/2019` as the `DATE`.

- `event Late Night Study Session /start 2200 /end 0100 /date 02/10/2019`

  Adds an **event** with `Late Night Study Session` as `DESCRIPTION`, `2200` as the `START_TIME`, `0100` as the `END_TIME` and `02/10/2019` as the `DATE`.

  Note that in this case, the absolute value of the `END_TIME` is ***not after*** the absolute value of the `START_TIME`. **COMPal** interprets this as your Late Night Study Session starting at 2200 on 02/10/2019, and ending at 0100 on 03/10/2019 (the following day).

⚠	An **event** has a maximum duration of 24 hours, i.e. if you enter the same value for both `START_TIME` and `END_TIME`, **COMPal** interprets it as a 24-hour long event. E.g. `event LAN Party /start 2300 /end 2300 /date 02/10/2019` means that your LAN Party starts at 2300 on 02/10/2019 and ends at 2300 on 03/10/2019 (the following day).

However, the above is merely the ***basic format***. As students, we have to juggle schoolwork, friends, family, and perhaps even a side job. As our lives become increasingly hectic, we invariably have to pick some **events** to prioritise above others. To handle these concerns, **COMPal** lets you assign ***priorities*** and also create ***recurring*** **events**.

4.2.2.1.1. Assigning Priorities to Events

The optional `/priority` keyword lets you assign an **event** with a `PRIORITY`. If you have an **event** that you ***absolutely have*** to complete, you can enter a `PRIORITY` of `high`. If your **event** isn't that urgent, you can use the value of `medium`, or if it isn't something worth worrying about, you can assign it as `low`.

Alternatively, if you do not use the `/priority` keyword, **COMPal** will set the `PRIORITY` of your **event** as `low` by ***default***.

Command Format (Priority):

```
event DESCRIPTION /start START_TIME /end END_TIME /date DATE /priority
PRIORITY
```

Examples:

- `event Netflix and Chill /start 2300 /end 0200 /date 02/10/2019 /priority high`

  Adds an **event** with `Netflix and Chill` as `DESCRIPTION`, `2300` as the `START_TIME`, `0200` as the `END_TIME`, `02/10/2019` as the `DATE`, and `high` as `PRIORITY`**.**

- `event Birthday Bash /start 1800 /end 2300 /date 02/10/2019 /priority medium`

  Adds an **event** with `Birthday Bash` as `DESCRIPTION`, `1800` as the `START_TIME`, `2300` as the `END_TIME`, `02/10/2019` as the `DATE`, and `medium` as `PRIORITY`**.**

- `event Study Session /start 1900 /end 2300 /date 02/10/2019 /priority low`

  Adds an **event** with `Study Session` as `DESCRIPTION`, `1900` as the `START_TIME`, `2300` as the `END_TIME`, `02/10/2019` as the `DATE`, and `low` as `PRIORITY`**.**

- `event Study Session /start 1900 /end 2300 /date 02/10/2019`

  This adds an **event** with `Study Session` as `DESCRIPTION`, `1900` as the `START_TIME`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`.

  However, omitting the `/priority` keyword prompts **COMPal** to automatically assign `low` as `PRIORITY`, meaning that the result of this command is ***identical*** to the previous command. This example illustrates the ***optional*** nature of the `/priority` keyword.

4.2.2.1.2. Adding Recurring Events

The optional `/final-date` and `/interval` keywords work in tandem to let you add multiple ***recurring*** **events**.

`/final-date` allows you to recur **events** until a specified `FINAL_DATE`, which is the ***final possible date*** for your final **event**. If you do not use the `/final-date` keyword, **COMPal** assumes you are not adding ***recurring*** **events**.

`/interval` allows you to specify the `INTERVAL` between each recurring **event**, in terms of number of days. If you do not use the `/interval` keyword, **COMPal** will set your `INTERVAL` to `7` by ***default*** i.e. every **event** occurs exactly ***seven days / one week*** after the previous, as long as it is not after the `FINAL_DATE` specified using the `/final-date` keyword.

Of course, in life, there's also a chance of ***regular, but not identical*** intervals between **events**. **COMPal** introduces a neat little hack: if you add multiple `DATE`s after your `/date` keyword, **COMPal** will add recurring **events** for each `DATE` using the `INTERVAL` (specified or default), as long as it is not after the `FINAL_DATE` specified using the `/final-date` keyword.

Command Format (Recurring):

```
event DESCRIPTION /start START_TIME /end END_TIME /date DATE /final-date
FINAL_DATE /interval INTERVAL
```

ℹ	You can use the `/priority` keyword here as well. Every ***recurring*** **event** will have the same `PRIORITY`.

Examples:

- `event Driving Lesson /start 1130 /end 1330 /date 02/10/2019`

  Adds an **event** with `Driving Lesson` as `DESCRIPTION`, `1130` as the `START_TIME`, `1330` as the `END_TIME`, `02/10/2019` as the `DATE`**.**

  This is the command with a ***basic format***. The following examples will show how adding `/final-date` and `/interval` changes the result.

- `event Driving Lesson /start 1130 /end 1330 /date 02/10/2019 /final-date 15/12/2019`

  Adds an **event** with `Driving Lesson` as `DESCRIPTION`, `1130` as the `START_TIME`, `1330` as the `END_TIME`, `02/10/2019` as the `DATE`, and `15/12/2019` as the `FINAL_DATE`. Omitting the `/interval` keyword means that your `INTERVAL` is set to `7` by ***default***. This means that:

1. 1130 to 1330, 02/10/2019, Wednesday is the ***first*** **event** when you have your first Driving Lesson.
2. **COMPal** then reminds you of a Driving Lesson for ***every*** Wednesday after that, until the last Wednesday before 15/12/2019 (which is 11/12/2019).

- `event Driving Lesson /start 1130 /end 1330 /date 02/10/2019 /final-date 15/12/2019 /interval 14`

  Adds an **event** with `Driving Lesson` as `DESCRIPTION`, `1130` as the `START_TIME`, `1330` as the `END_TIME`, `02/10/2019` as the `DATE`**,** `15/12/2019` as the `FINAL_DATE`, and `14` as the `INTERVAL`. This means that:

  3. 1130 to 1330, 02/10/2019, Wednesday is the ***first*** **event** when you have your first Driving Lesson.
  4. **COMPal** then reminds you of a Driving Lesson for ***every alternate*** Wednesday (***two weeks*** after the previous Wednesday), until the last ***alternate*** Wednesday before 15/12/2019 (which is 27/11/2019).

- `event Driving Lesson /start 1130 /end 1330 /date 02/10/2019 04/10/2019 /final-date 15/12/2019 /interval 14`

  Adds an **event** with `Driving Lesson` as `DESCRIPTION`, `1130` as the `START_TIME`, `1330` as the `END_TIME`, `02/10/2019` and `04/10/2019` as the `DATE`**,** `15/12/2019` as the `FINAL_DATE`, and `14` as the `INTERVAL`. This means that:

  3. 1130 to 1330, 02/10/2019, Wednesday and 1130 to 1330, 04/10/2019, Friday are the ***first*** **event** ***iterations*** where you have your first few Driving Lessons.
  4. **COMPal** then reminds you of a Driving Lesson for ***every alternate*** Wednesday (***two weeks*** after the previous Wednesday) and ***every alternate*** Friday (***two weeks*** after the previous Friday), until the last ***alternate*** Friday before 15/12/2019 (which is 29/11/2019).

- `event Driving Lesson /start 1130 /end 1330 /date 02/10/2019 /final-date 15/12/2019 /priority high`

  Adds an **event** with `Driving Lesson` as `DESCRIPTION`, `1130` as the `START_TIME`, `2359` as the `END_TIME`, `02/10/2019` as the `DATE`, `15/12/2019` as the `FINAL_DATE`, and `high` as the `PRIORITY`. This means that:

  3. 1130 to 1330, 02/10/2019, Wednesday is the ***first*** **event** when you have your first Driving Lesson.
  4. **COMPal** then reminds you of a Driving Lesson for ***every*** Wednesday after that, until the last Wednesday before 15/12/2019 (which is 11/12/2019). For ***every*** **event**, the `PRIORITY` is `high`.

💡	Adding multiple `DATE`s without using `/final-date` will still add multiple **events** with the specified `DATE`s. This may be helpful if your **events** are extremely irregular, and somehow you have outrageous photographic memory and remember the exact dates of all your **events**.

⚠	If you add multiple `DATE`s, ensure that your `FINAL-DATE` is after your latest `DATE`, otherwise, any **event** with a `DATE` after your `FINAL-DATE` will not be added.

Full Command Format (With all keywords and parameters):

```
event DESCRIPTION /start START_TIME /end END_TIME /date DATE /final-date
FINAL_DATE /interval INTERVAL /priority PRIORITY
```

💡	Did you know that you do not have to follow the order in the format above? **COMPal** is able to detect the keywords as long as you enter them.

⚠	Despite the above tip, do note that your `DESCRIPTION` ***MUST*** be after your very first command word, i.e. `event`.
⚠	Take care not to enter any `/` into the command box, as **COMPal** may mistake it as an ***invalid parameter***!!!!

4.2.2.2. Editing Events: `edit`

Need to change information about an **event**? Enter `edit /id TASK_ID` along with the `FIELD` to edit, where TASK_ID is the **task's** id number.

Command Format:

- Editing `DESCRIPTION`: `edit /id TASK_ID /description NEW_DESCRIPTION`
- Editing `DATE`: `edit /id TASK_ID /date NEW_DATE`
- Editing `PRIORITY`: `edit /id TASK_ID /priority NEW_PRIORITY`
- Editing `START_TIME`: `edit /id TASK_ID /end NEW_START_TIME`
- Editing `END_TIME`: `edit /id TASK_ID /end NEW_END_TIME`

You can edit more than one parameter with a single command.

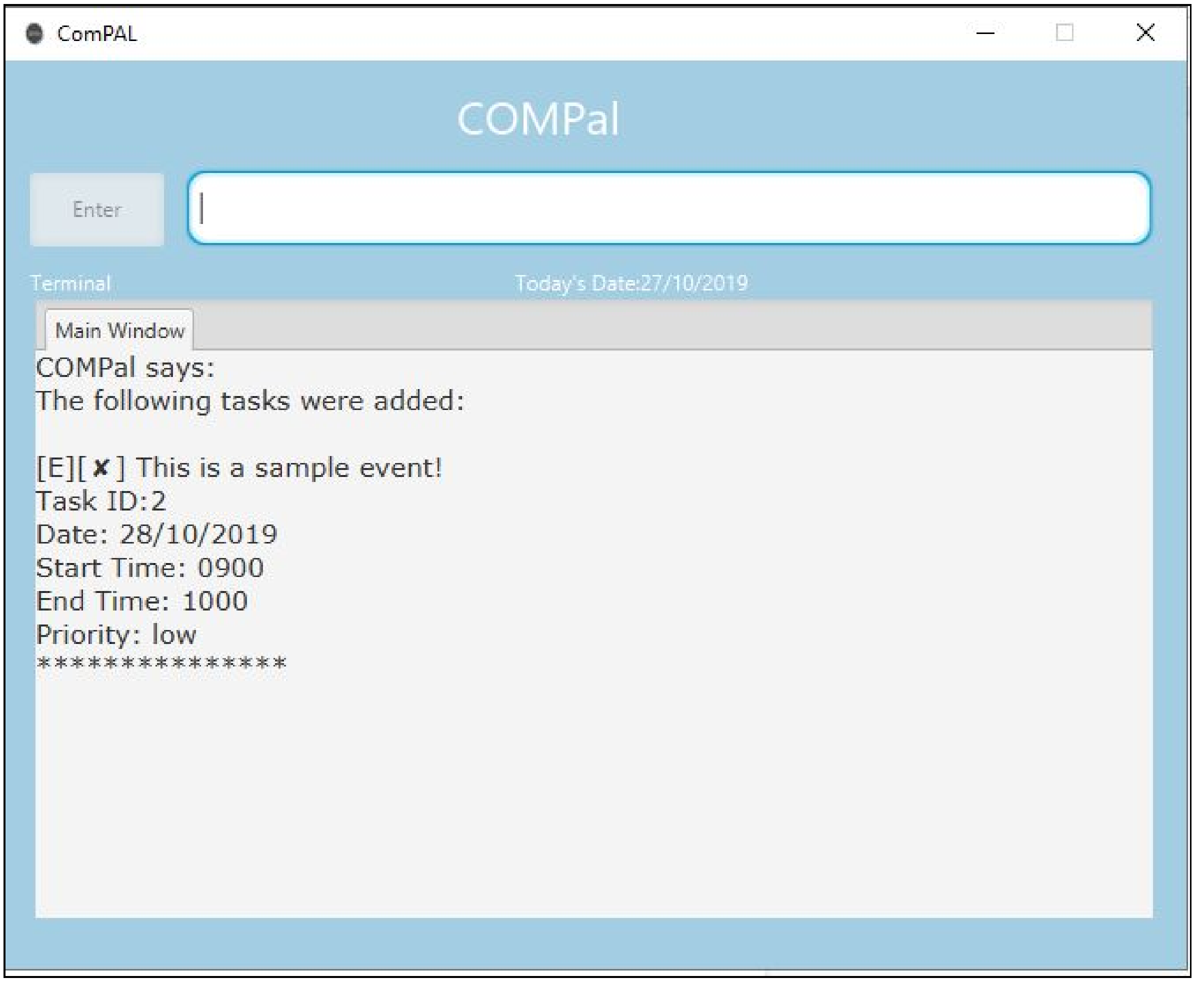
Full Command Format (including all possible keywords):

```
edit /id TASK_ID /description NEW_DESCRIPTION /start NEW_START_TIME /end
NEW_END_TIME /date NEW_DATE /priority NEW_PRIORITY
```

⚠	You ***CANNOT*** change `FINAL-DATE` or `INTERVAL`. These parameters are only for ***adding multiple*** **events**, they are not properties of each **event**. You can, however, delete any **event** that you think is not necessary anymore, or edit each **event**.

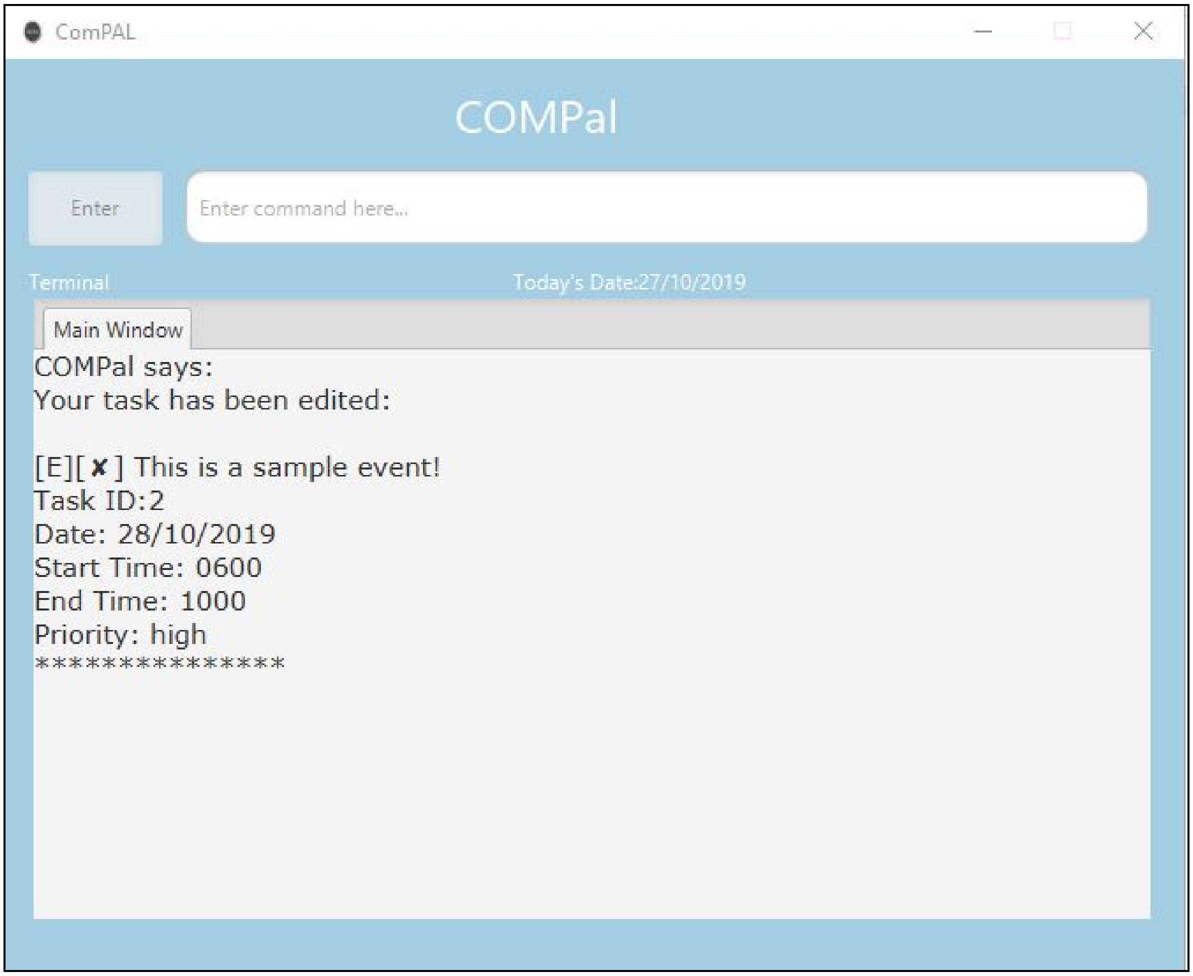
For example, if you wish to increase the priority of an **event** with an id number of 2 currently set to low and bring forward its starting time to 6 am, do:

```
edit /id 2 /priority high /start 0600
```



```
COMPal says:
The following tasks were added:

[E][✘] This is a sample event!
Task ID:2
Date: 28/10/2019
Start Time: 0900
End Time: 1000
Priority: low
***************
```

Figure 17. The original **event** with `TASK_ID` of `2` before the `edit` command executes



```
COMPal says:
Your task has been edited:

[E][✘] This is a sample event!
Task ID:2
Date: 28/10/2019
Start Time: 0600
End Time: 1000
Priority: high
***************
```

Figure 18. The **event** with `TASK_ID` of `2` after the `edit` command executes

Note that `PRIORITY` is changed from `low` as shown in Figure 16 to `high` in Figure 17, and `START_TIME` has been changed from `0900` to `0600,` as intended.

💡	**COMPal** stores your **events** in a `tasks.txt` file, in clear text form. If you are curious by nature and love looking at raw data, you can edit your **events** directly in the `tasks.txt` file.

⚠	While you can edit **events** directly and show your friends that you “hacked” **COMPal**, you cannot add **events** directly in the `tasks.txt` file. You can only use the `event` command to add **events**.
⚠	If you read `tasks.txt` closely, you may notice that the various parameters of each **event** (such as `DESCRIPTION`, `PRIORITY`, etc.) are separated by underscores (`_`). ***Do not remove*** these underscores and ensure that each parameter corresponds to the correct type (e.g. `PRIORITY` must always be `low`, `medium`, or `low`, etc.), otherwise

	COMPal will display undefined behaviour.

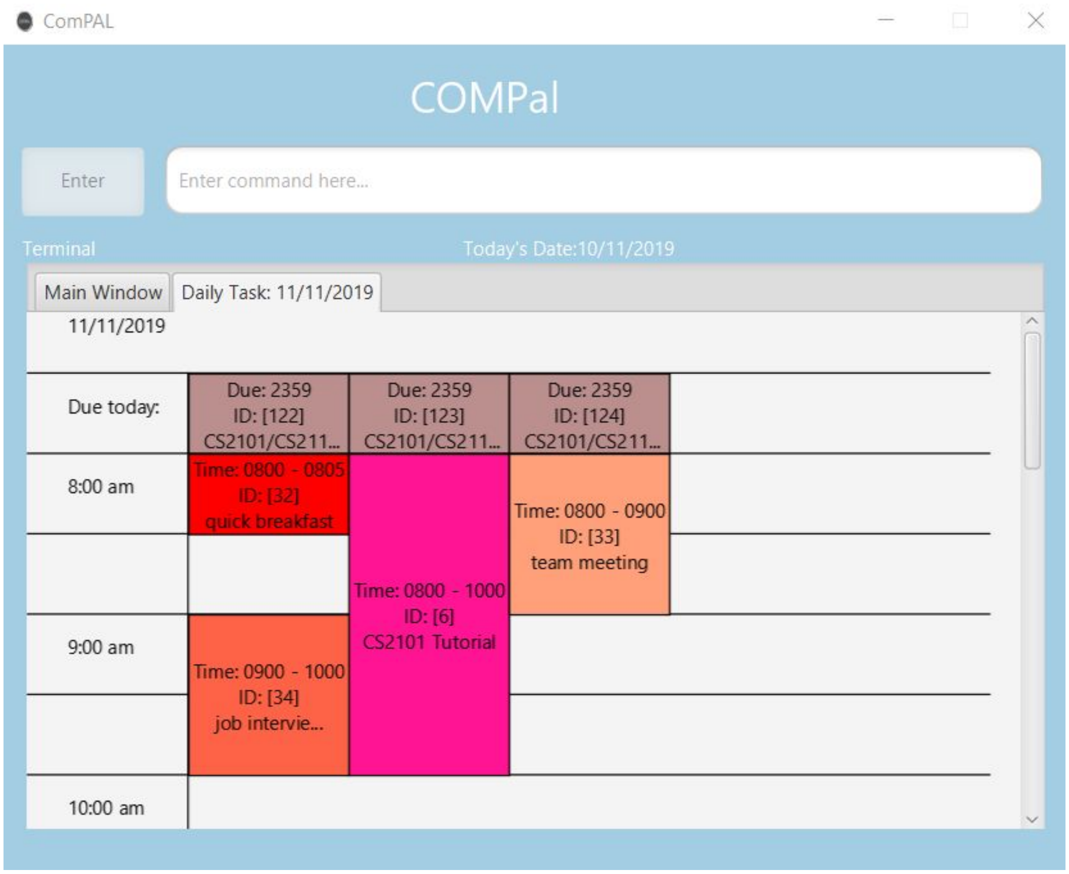
4.3. Quick overview of the generated daily schedule.

After you have filled COMPal with **tasks** from your hectic schedule, you can use the `view day` command to look at your generated daily schedule.

The quick and intuitive interface enables you to quickly understand what deadlines are due and what clashing events you may have! Fret not as COMPal has sorted what is important and ***shows you the important tasks at hand that are not completed!*** Additionally, the more important **tasks** are placed on the left while the less-important one is to the right!

You ***should*** view the full information on the viewed **task** in the `main window`! As the daily **task** serves as a quick reference for you!

⚠	The Daily Task tab is static! Meaning you need to invoke `view day` to refresh your updated calendar if you were to add a task!
ℹ	Only the ***5 most important tasks*** will be ***displayed*** for each ***hourly slot*** to allow you to ***focus on the tasks with higher priority***!



11/11/2019

Due today:
Due: 2359 ID: [122] CS2101/CS211...
Due: 2359 ID: [123] CS2101/CS211...
Due: 2359 ID: [124] CS2101/CS211...

8:00 am
Time: 0800 - 0805 ID: [32] quick breakfast
Time: 0800 - 1000 ID: [6] CS2101 Tutorial
Time: 0800 - 0900 ID: [33] team meeting

9:00 am
Time: 0900 - 1000 ID: [34] job intervie...

10:00 am

Figure 19. Generated planner for `view day` command in `Daily Task: DATE` tab. The most left schedule for each hourly slot represents the most important **task** to be completed first!

ℹ	The below highlights are the color-coded type of **task** that can be seen when viewing COMPal: Non-school related event: High, medium, low priority School-related event: Lecture, Tutorial, Sectional or lab Task stored as deadline: deadline

5. Future Enhancements

The current implementation of **COMPal** follows the paradigm of extensibility and flexibility. It is very easy to add and/or refine the currently available features. Here is a list of some possible enhancements that may come with **COMPal** v2.0:

1. **COMPal** will advise you on clashes and even potentially better slots (based on a ‘busyness’ index that **COMPal** will calculate from your current schedule)
    a. This will involve **COMPal** detecting clashes and finding a new free time slot for you. Finding free time slots is already a feature implemented in **COMPal.**
2. **COMPal** will have pop-up reminders instead of reminders being a command
    a. This will involve **COMPal** running a background ‘checker’ (thread) to check if there is a reminder set for a certain **task** and will then remind you with a pop-up window if necessary and when appropriate
3. **COMPal** chatbot
    a. **COMPal** will no longer only take in commands, but can also understand normal speech.
    b. For example, you can enter ‘What do I have on today?’ and **COMPal** will understand it as a command to view the day’s **tasks** and so will display it to you
4. **COMPal** allows you to share your schedule with your friends
    a. The ability to share your schedule with your friends, to find free time to hang out together, is understandably important
    b. With this enhancement, you can view the schedules of your friends and organize your schedule together

6. Frequently Asked Questions

Q: How do I transfer my current schedule data to another computer?

A: Now with the export/import function, you can export your stored schedule into an iCalendar file and move the file to the new computer so you can import your previous schedule from it. (Remember to download COMPal on the other computer though!)

Q: How do I transfer my schedule to another scheduling application (e.g google calendar)?

A: Most scheduling application has the ability to import your stored schedule from an iCalendar file! Here are some links that might help you get started:
1. iCalender files into outlook
2. iCalender files into google calendar

Q: Am I able to transfer my schedule from another application(e.g google calendar) to COMPal?

A: Yes you can! However take note that by default, the priority of the imported events is set to low! You can `edit` it to set the priority to your preference after importing it

7. Command Summary

ℹ Command Format

- Words in `UPPER_CASE` are the parameters to be supplied by the user e.g. `delete TASK_ID`, `TASK_ID` is a parameter which can be used as `delete 5`.
- Items in square brackets are optional e.g. `view day [/date DATE] [/type TASK_TYPE]` can be used as `view day /date 29/10/2019` or as `view day`.
- Parameters can be in any order e.g. if the command specifies `/id TASK_ID /status Y/N`, `/status Y/N /id TASK_ID` is also acceptable.
- Items in angle brackets are optional, but at least one item must be included e.g. `</description DESCRIPTION> </date DATE>` can be used as `/description go for a run` or `/description go for a run /date 28/10/2019`

- Help: `help`
  - List all commands: `help`
  - View a command by name: `help /search COMMAND_NAME`
    e.g. `help /search edit`
- Delete: `delete TASK_ID`
  e.g. `delete 2`
- Done: `done /id TASK_ID /status Y/N`
  e.g. `done /id 5 /status Y` or `done /id 3 /status N`
- Find: `find KEYWORD`
  e.g. `find cs2113t`
- View: `view`
  - Day: `view day [/date DATE] [/type TASK_TYPE]`
    e.g. `view day` or `view day /date 22/10/2019 /type deadline`
  - Week: `view week [/date DATE] [/type TASK_TYPE]`
    e.g. `view week /date 28/10/2019`
  - Month: `view month [/date DATE] [/type TASK_TYPE]`
    e.g. `view month /type event`
- List:
  - List all **tasks**: `list`
  - List specific **task** type: `list /type TASK_TYPE`
    e.g. `list /type event`
- View reminder: `view-reminder`
- Set reminder: `set-reminder /id TASK_ID /status Y/N`
  e.g. `set-reminder /id 1 /status Y`

- Find free slot: `findfreeslot /date DATE /hour HOUR /min MINUTE`
  e.g. `findfreeslot /date 28/10/2019 /hour 4 /min 0`
- Edit: `edit /id TASK_ID </description TASK_DESCRIPTION> </date DATE> </priority PRIORITY> </start START_TIME> </end END_TIME>`
  e.g. `edit /id 5 /date 26/10/2019 /priority high` or `edit /id 1 /description go for a run`
- Bye: `bye`
- Deadline: `deadline TASK_DESCRIPTION /date DATE /end END_TIME [/priority PRIORITY] [/final-date FINAL_DATE]`
  e.g. `deadline CS2100 assignment /date 29/10/2019 /end 1200 /priority high`
- Event: `event TASK_DESCRIPTION /date DATE /start START_TIME /end END_TIME [/priority PRIORITY] [/final-date FINAL_DATE]`
  e.g. `event CS2101 lesson /date 21/10/2019 /start 0800 /end 1000`