AB-3

User Guide

NUS Module Planner is a **desktop app for NUS students to manage and plan the modules to enrol in upcoming semesters, optimized for use via a Command Line Interface** (CLI) while still having the benefits of a Graphical User Interface (GUI). If you can type fast, NUS Module Planner can get your module planning done faster than traditional GUI apps.

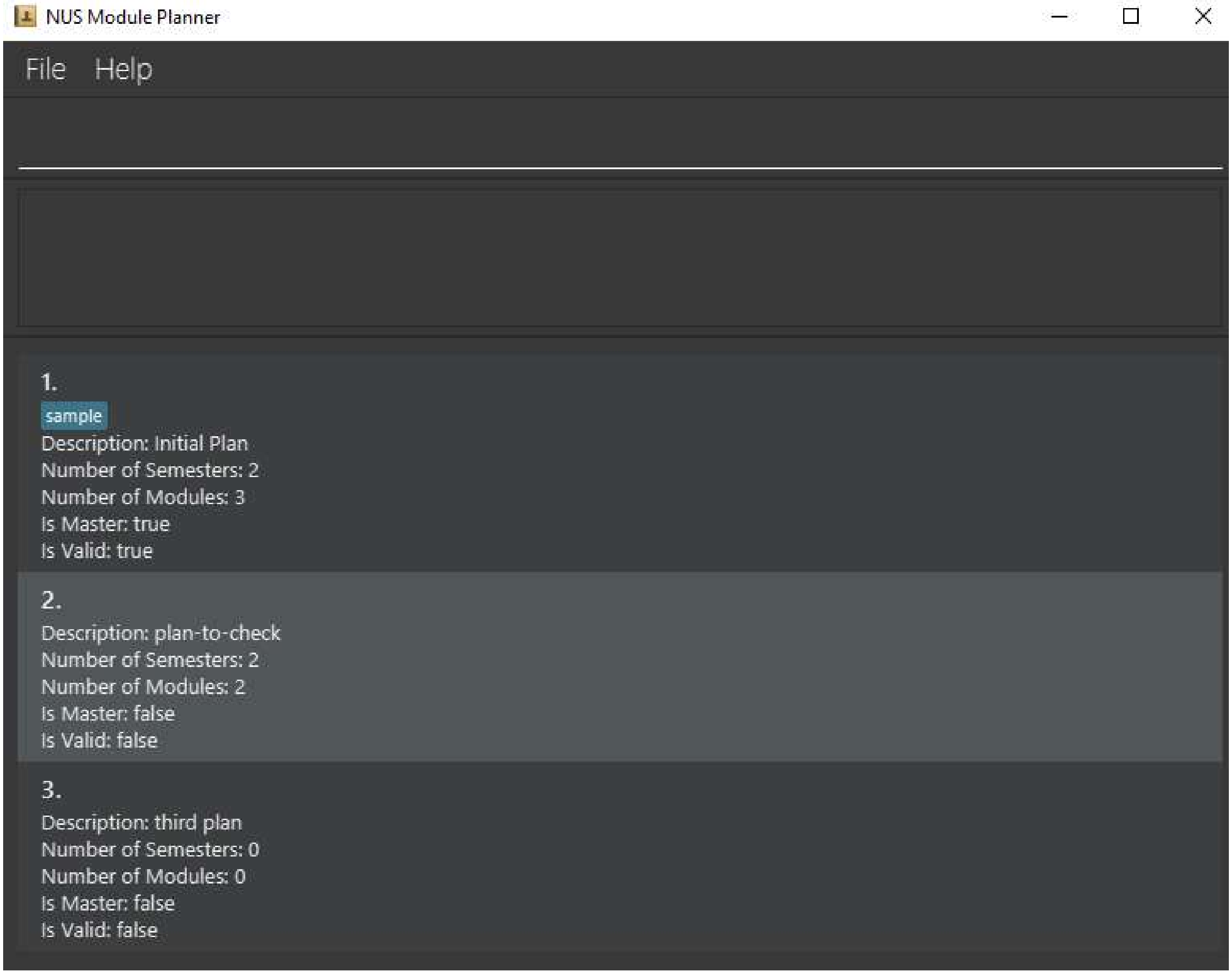
Quick start

1. Ensure you have Java `11` or above installed in your Computer.

2. Download the latest `moduleplanner.jar` from here.

3. Copy the file to the folder you want to use as the *home folder* for your AddressBook.

4. Double-click the file to start the app. The GUI similar to the below should appear in a few seconds. Note how the app contains some sample data.



1.
sample
Description: Initial Plan
Number of Semesters: 2
Number of Modules: 3
Is Master: true
Is Valid: true

2.
Description: plan-to-check
Number of Semesters: 2
Number of Modules: 2
Is Master: false
Is Valid: false

3.
Description: third plan
Number of Semesters: 0
Number of Modules: 0
Is Master: false
Is Valid: false

5. Type the command in the command box and press Enter to execute it. e.g. typing `help` and pressing Enter will open the help window.
   Some example commands you can try:

   - `list` : Lists all plans.
   - `addp d/sample plan` : Adds a plan with description `sample plan` to the Address Book.
   - `deletep p/3` : Deletes the 3rd plan shown in the current list.
   - `clear` : Deletes all plans.
   - `exit` : Exits the app.

6. Refer to the Features below for details of each command.

Features

Notes about the command format:

- Words in `UPPER_CASE` are the parameters to be supplied by the user.
  e.g. in `addp d/DESCRIPTION`, `DESCRIPTION` is a parameter which can be used as `addp d/Networking Focus Area`.

- Items in square brackets are optional.
  e.g `d/planDesc [t/TAG]` can be used as `d/planDesc t/3 years`.

- Items with `…` after them can be used multiple times including zero times.
  e.g. `[t/TAG]…` can be used as ` ` (i.e. 0 times), `t/3 years`, `t/3 years t/2 internships` etc.

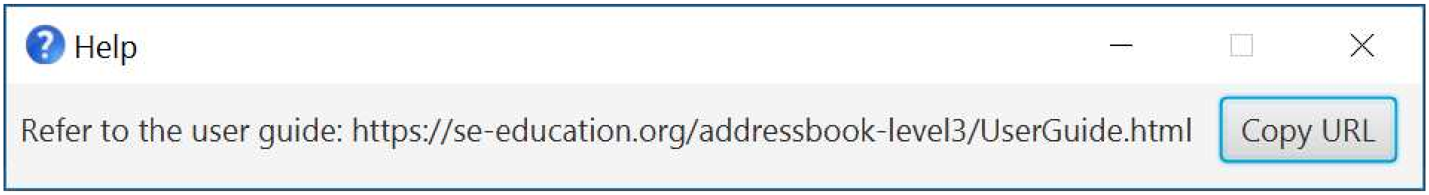
- Parameters can be in any order.
  e.g. if the command specifies `p/PLAN_NUMBER s/SEM_NUMBER`, `s/SEM_NUMBER p/PLAN_NUMBER` is also acceptable.

- If a parameter is expected only once in the command but you specified it multiple times, only the last occurrence of the parameter will be taken.
  e.g. if you specify `d/myplan1 d/myplan2`, only `d/myplan2` will be taken.

- Extraneous parameters for commands that do not take in parameters (such as `help`, `list`, `exit` and `clear`) will be ignored.
  e.g. if the command specifies `help 123`, it will be interpreted as `help`.

Viewing help : `help`

Shows a message explaining how to access the help page.



Refer to the user guide: https://se-education.org/addressbook-level3/UserGuide.html

Format: `help`

User Commands

General Commands

Check graduation : `validate`

Format: `validate`

> *Note: The validate command checks if a plan is valid by looking at whether or not the modules contained in its history match those of the current `master`. Hence, `master` is always valid. The command does not check for pre-requisites and preclusions with this release.*

> *Example: if master is at current semester `2`, all other plans have their semester `1`s checked, if these semester `1`s match the master module, the plan is valid.*

- Shows header
  - Shows plan number
  - Shows description of plan
  - Shows if the plan is valid compared to master plan
  - Shows how many MCs the plan has
  - Shows how many semesters the plan has

Clearing all entries : `clear`

Clears all entries from the address book.

Format: `clear`

Exiting the program : `exit`

Exits the program.

Format: `exit`

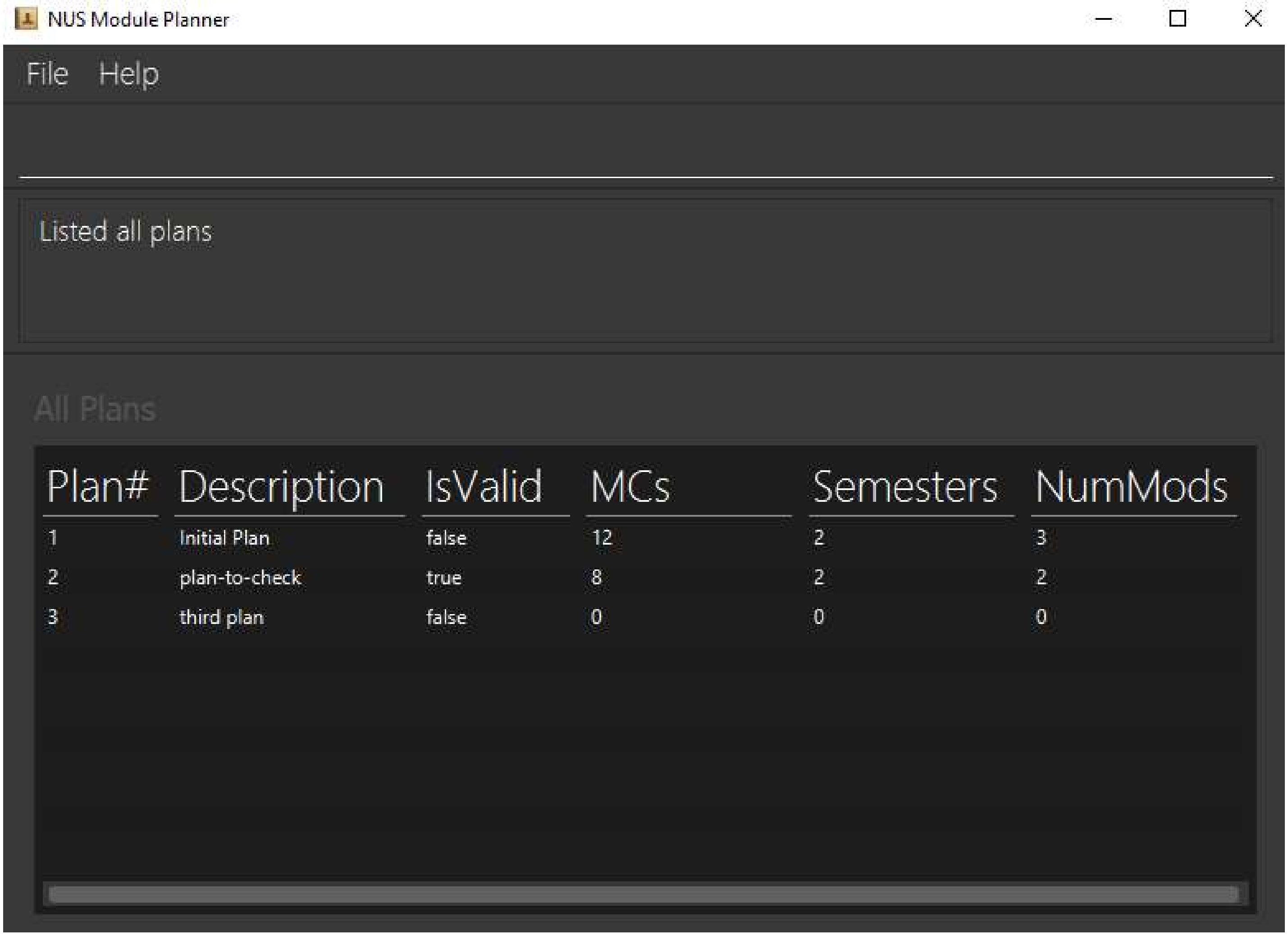
Plan commands

List a summary of all plans: `list`

Format: `list`

This command must be done by the user at least once before they can use other commands. Marks the given plan as the master plan, and this plan should contain all the modules that the user has taken (if any).

Example output:



Listed all plans

All Plans

Plan#	Description	IsValid	MCs	Semesters	NumMods
1	Initial Plan	false	12	2	3
2	plan-to-check	true	8	2	2
3	third plan	false	0	0	0

Create/Delete Plan: `addp` / `deletep`

Format for adding: `addp d/DESCRIPTION [OPTIONAL: t/TAG...]`

Format for deleting: `deletep p/PLAN_NUMBER`

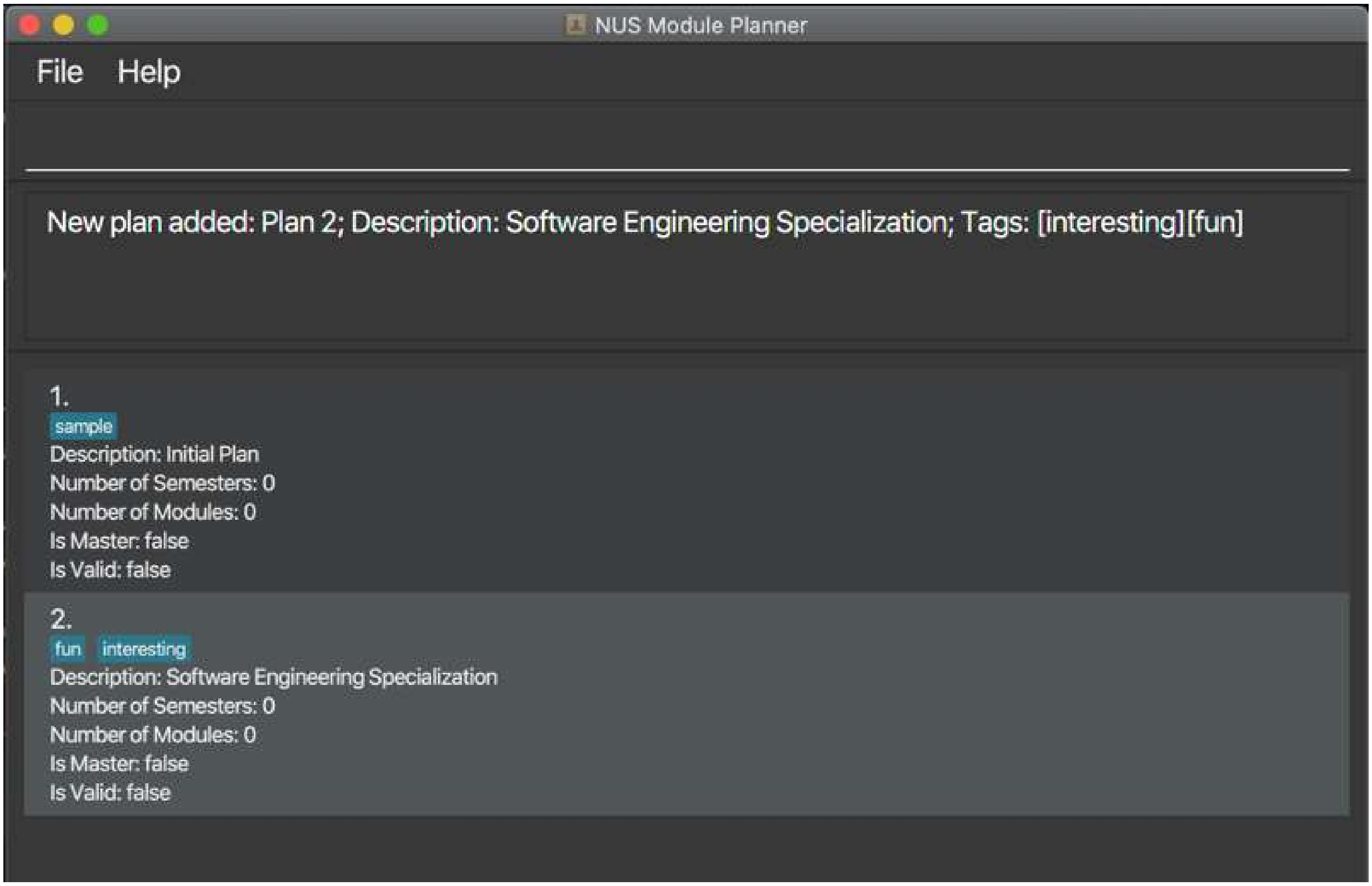
Shows 2 rows:

- Whether the plan is added/deleted successfully/unsuccessfully
- Plan number

Constraints:

- Trying to add a plan with invalid description will not be allowed
- Trying to delete a plan that does not exist will not be allowed
- Tags provided when adding a plan should contain only alphanumeric characters with no spaces

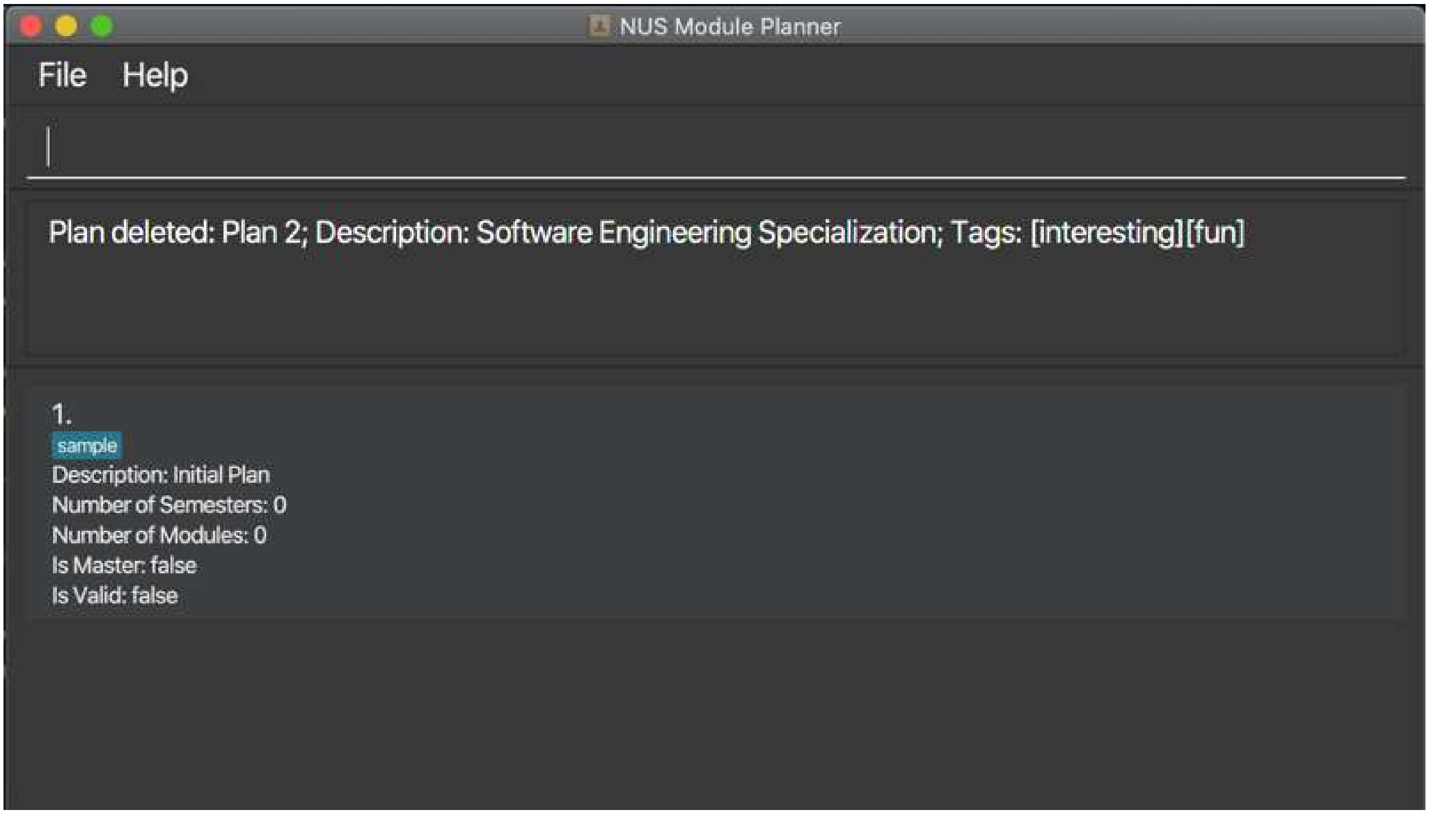
Example output for adding plan:



```
New plan added: Plan 2; Description: Software Engineering Specialization; Tags: [interesting][fun]
```

```
1.
sample
Description: Initial Plan
Number of Semesters: 0
Number of Modules: 0
Is Master: false
Is Valid: false
2.
fun interesting
Description: Software Engineering Specialization
Number of Semesters: 0
Number of Modules: 0
Is Master: false
Is Valid: false
```

Example output for deleting plan:



Plan deleted: Plan 2; Description: Software Engineering Specialization; Tags: [interesting][fun]

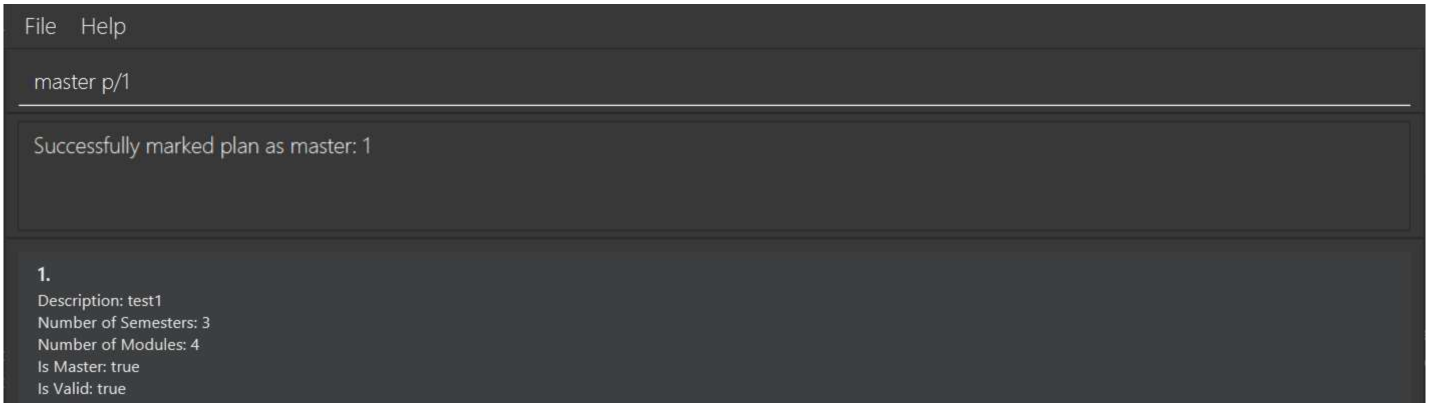
```
1.
sample
Description: Initial Plan
Number of Semesters: 0
Number of Modules: 0
Is Master: false
Is Valid: false
```

Set Plan as Master Plan: `master`

Format: `master p/PLAN_NUMBER`

This command must be done by the user at least once before they can use other commands. Marks the given plan as the master plan, and this plan should contain all the modules that the user has taken (if any).

Example output:



```
master p/1
```

Successfully marked plan as master: 1

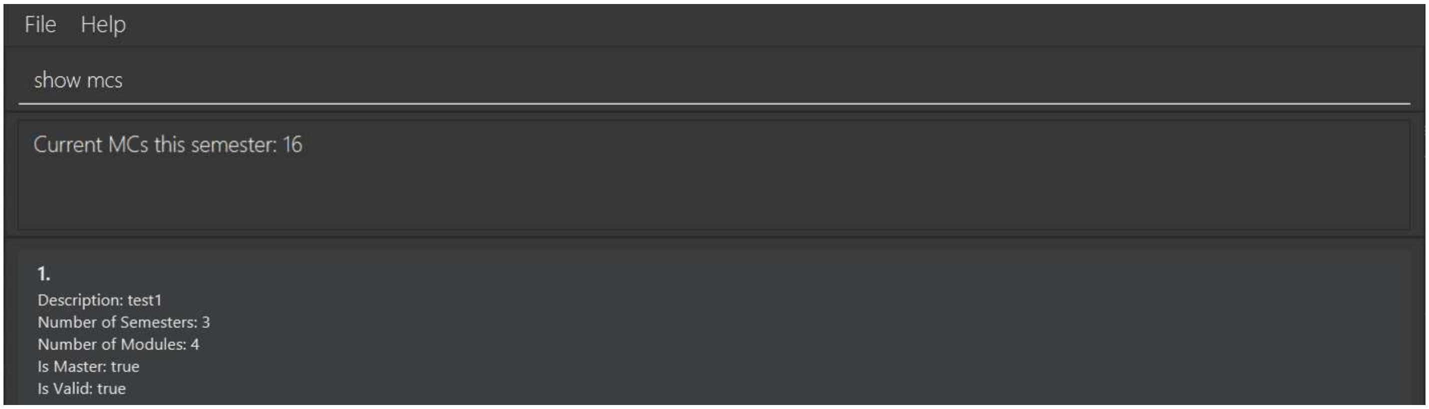
```
1.
Description: test1
Number of Semesters: 3
Number of Modules: 4
Is Master: true
Is Valid: true
```

Semester commands

Show the number of MCs the user is currently taking: `show MCs`

Format: `show mcs`

Example output:



```
show mcs
```

```
Current MCs this semester: 16
```

```
1.
Description: test1
Number of Semesters: 3
Number of Modules: 4
Is Master: true
Is Valid: true
```

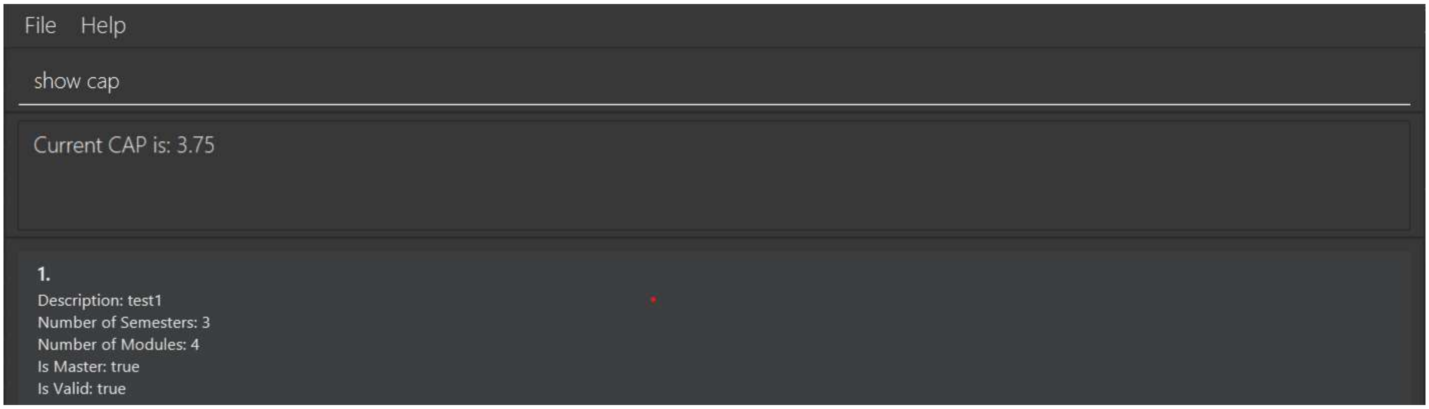
Calculate and show the current CAP (Cumulative academic points) of the student: `show cap`

Format: `show cap`

This command takes in the grades of modules user has marked as completed and entered their grade, and calculate their CAP using this formula:

$$\text{CAP} = \frac{\text{sum (module grade point x MCs assigned to module)}}{\text{sum (MCs assigned to all modules used in calculating the numerator)}}$$

Example output:



```
show cap
```

```
Current CAP is: 3.75
```

```
1.
Description: test1
Number of Semesters: 3
Number of Modules: 4
Is Master: true
Is Valid: true
```

Add/Delete Semester to/from Plan: `adds` / `deletes`

Format for adding: `adds p/PLAN_NUMBER s/SEM_NUMBER`

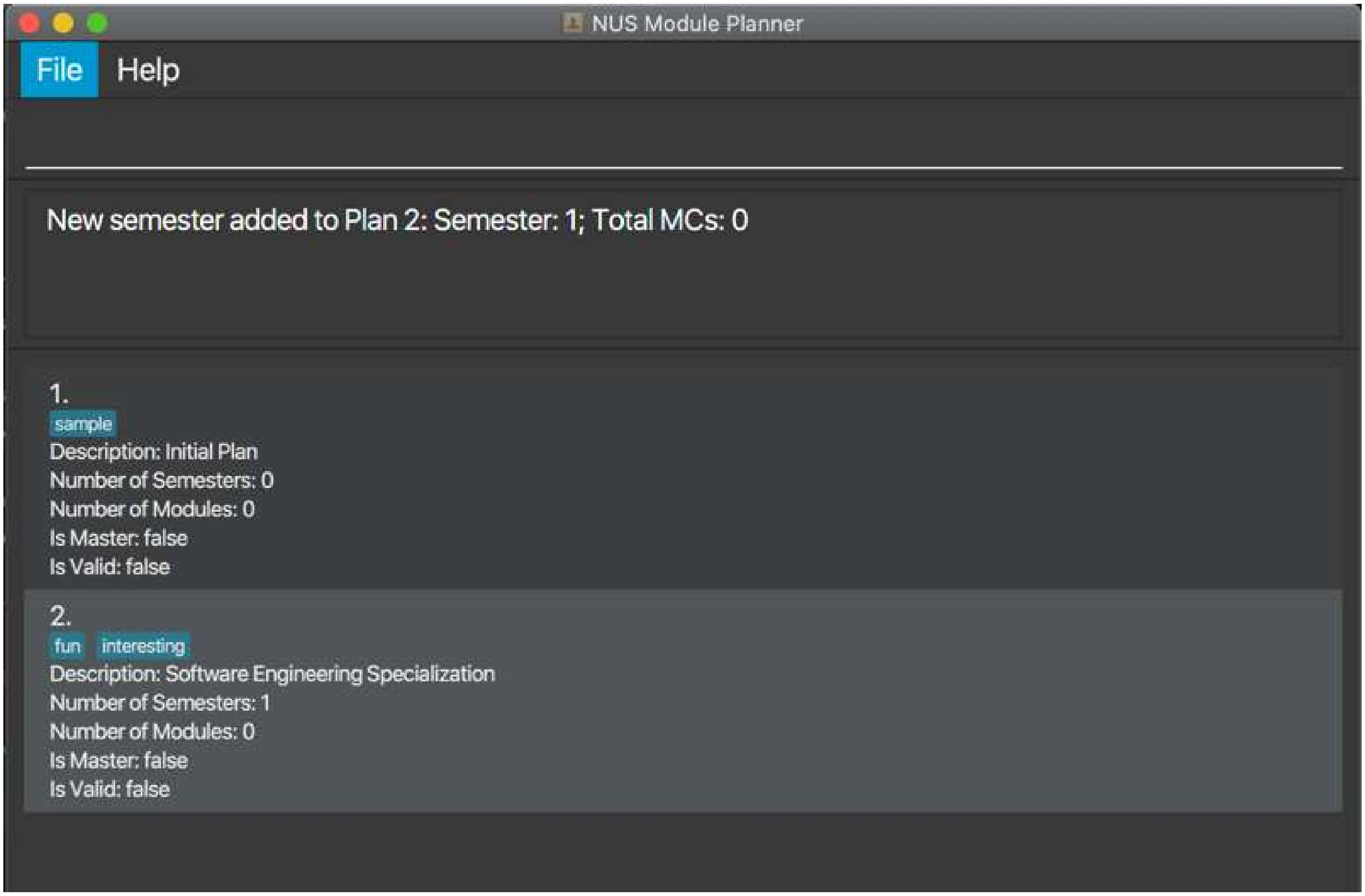
Format for deleting: `deletes p/PLAN_NUMBER s/SEM_NUMBER`

The output will show whether the operation was successful and include the semester number in its output.

Constraints:

- Trying to add a semester that already exist will not be allowed
- Trying to delete a semester that does not exist will not be allowed
- A semester number must be a positive integer

Example output for adding semester:

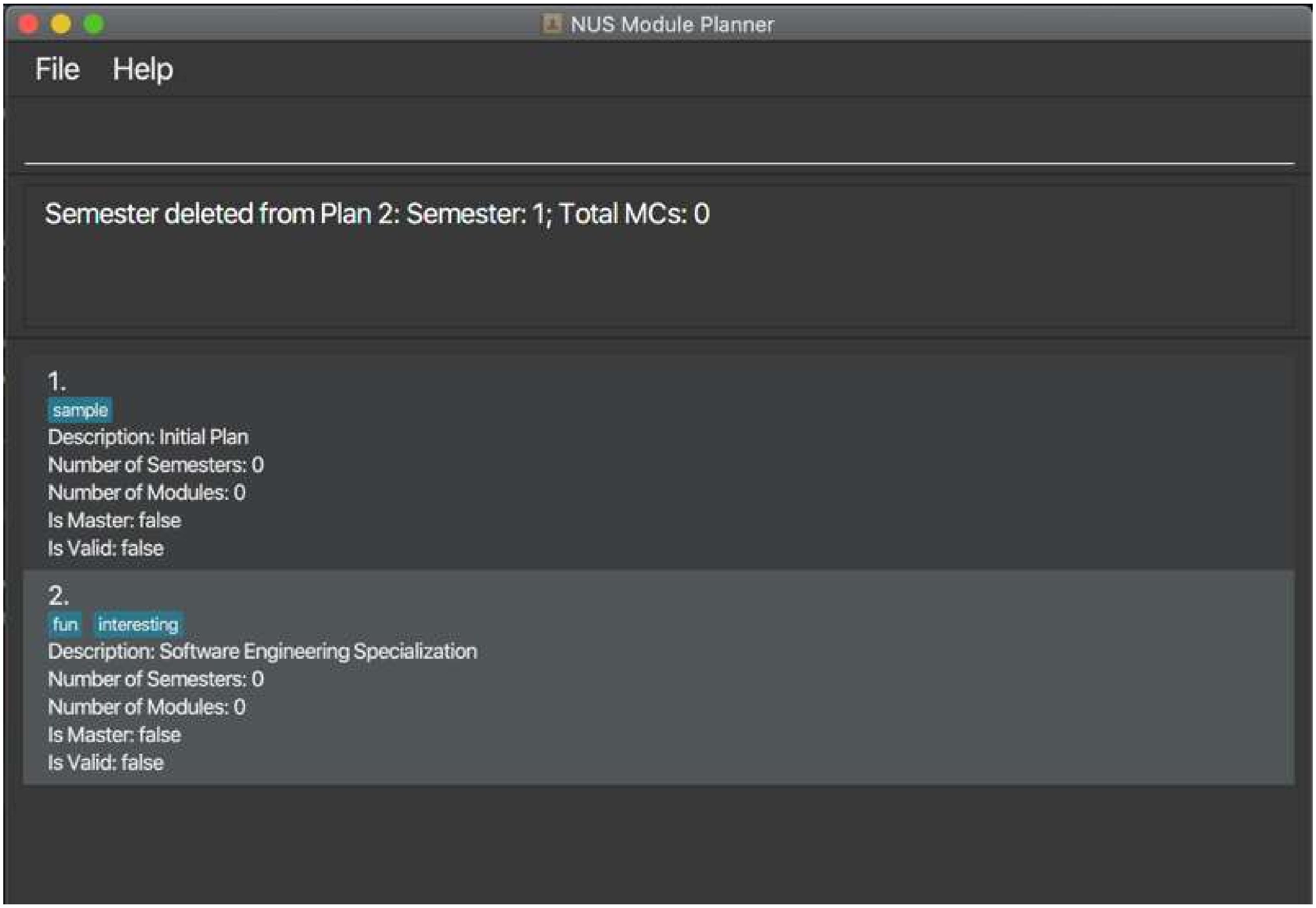


```
New semester added to Plan 2: Semester: 1; Total MCs: 0
```

1.
sample
Description: Initial Plan
Number of Semesters: 0
Number of Modules: 0
Is Master: false
Is Valid: false

2.
fun interesting
Description: Software Engineering Specialization
Number of Semesters: 1
Number of Modules: 0
Is Master: false
Is Valid: false

Example output for deleting semester:



Semester deleted from Plan 2: Semester: 1; Total MCs: 0

1.
sample
Description: Initial Plan
Number of Semesters: 0
Number of Modules: 0
Is Master: false
Is Valid: false

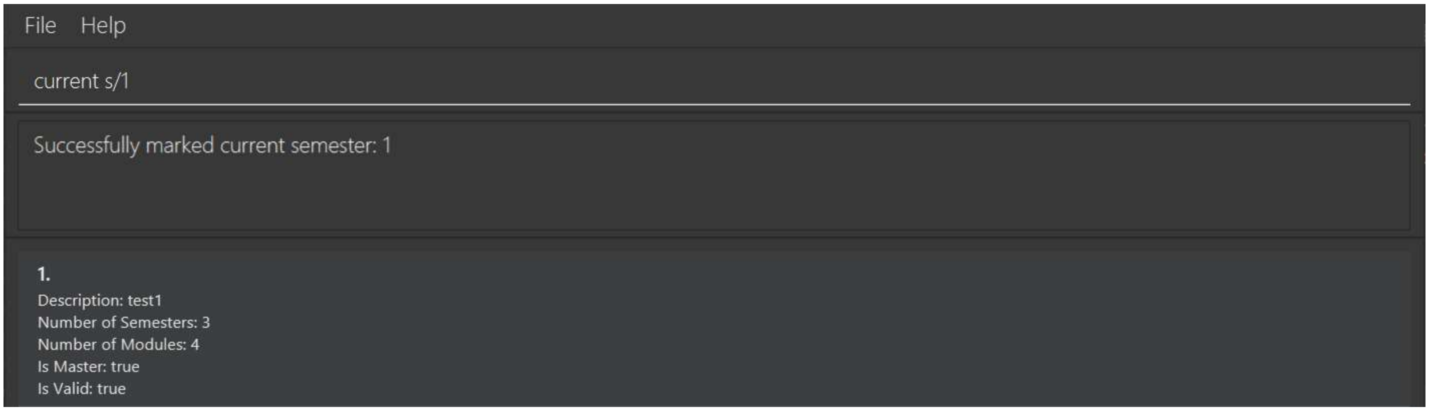
2.
fun interesting
Description: Software Engineering Specialization
Number of Semesters: 0
Number of Modules: 0
Is Master: false
Is Valid: false

Set Semester as in-progress: `current`

Format: `current s/SEM_NUMBER`

Marks the supplied semester as the current semester of the master plan. This indicates that all previous semesters are part of the user’s history and all future semesters have yet to be attempted. The user will have to manually update the current semester as time progresses.

Example output:



```
current s/1
```

Successfully marked current semester: 1

1.
Description: test1
Number of Semesters: 3
Number of Modules: 4
Is Master: true
Is Valid: true

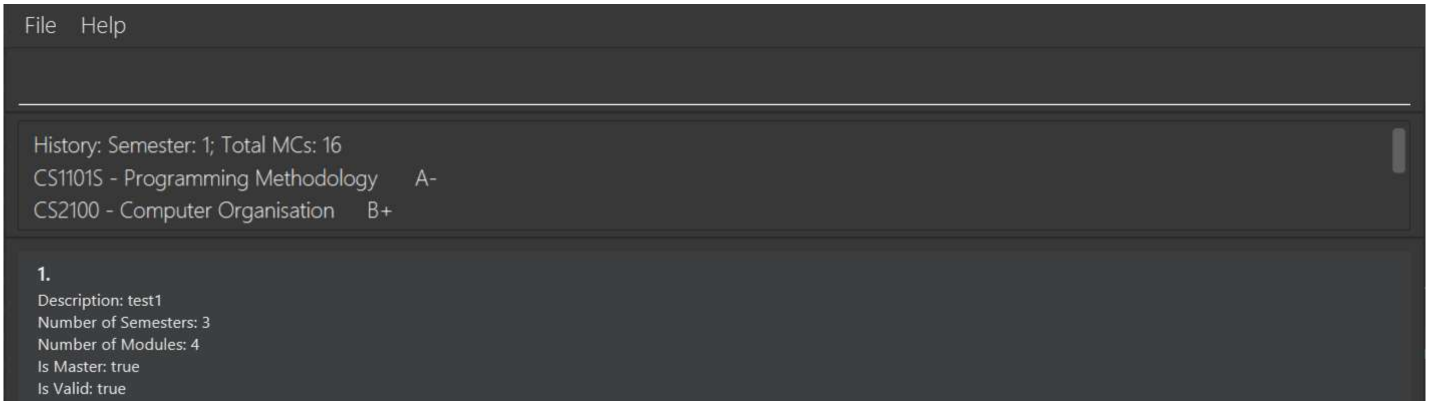
Show history: `history`

Format: `history`

The above command takes no arguments and shows the user a list of modules that they have completed up until **before** the *current semester*.

> *Tip: The* current semester *is the semester that was marked using the* `current semester` *command.*

Example output:



```
History: Semester: 1; Total MCs: 16
CS1101S - Programming Methodology    A-
CS2100 - Computer Organisation    B+
```

```
1.
Description: test1
Number of Semesters: 3
Number of Modules: 4
Is Master: true
Is Valid: true
```

Module commands

Add/Delete module to/from semester: `addm/deletem`

Format for adding: `addm m/MODULE_CODE p/PLAN_NUMBER s/SEM_NUMBER`

Format for adding module with grade: `addm m/MODULE_CODE p/PLAN_NUMBER s/SEM_NUMBER g/GRADE`

Format for deleting: `deletem m/MODULE_CODE p/PLAN_NUMBER s/SEM_NUMBER`

`SEM_NUMBER` here references to the semester number of the semester user created.

For example, if user added a semester using `adds p/1 s/5`, to add module to this semester user need to use `addm p/1 s/5 m/MODULE_CODE`

> *Tip: A user can view module info to see more details about it. (See* `info`*)*

This command takes in three arguments, `MODULE_CODE`, `PLAN_NUMBER` and `SEM_NUMBER`, and outputs meta details about the module being added/deleted, as well as whether the addition/deletion was successful or not.

The details to output are as follows:

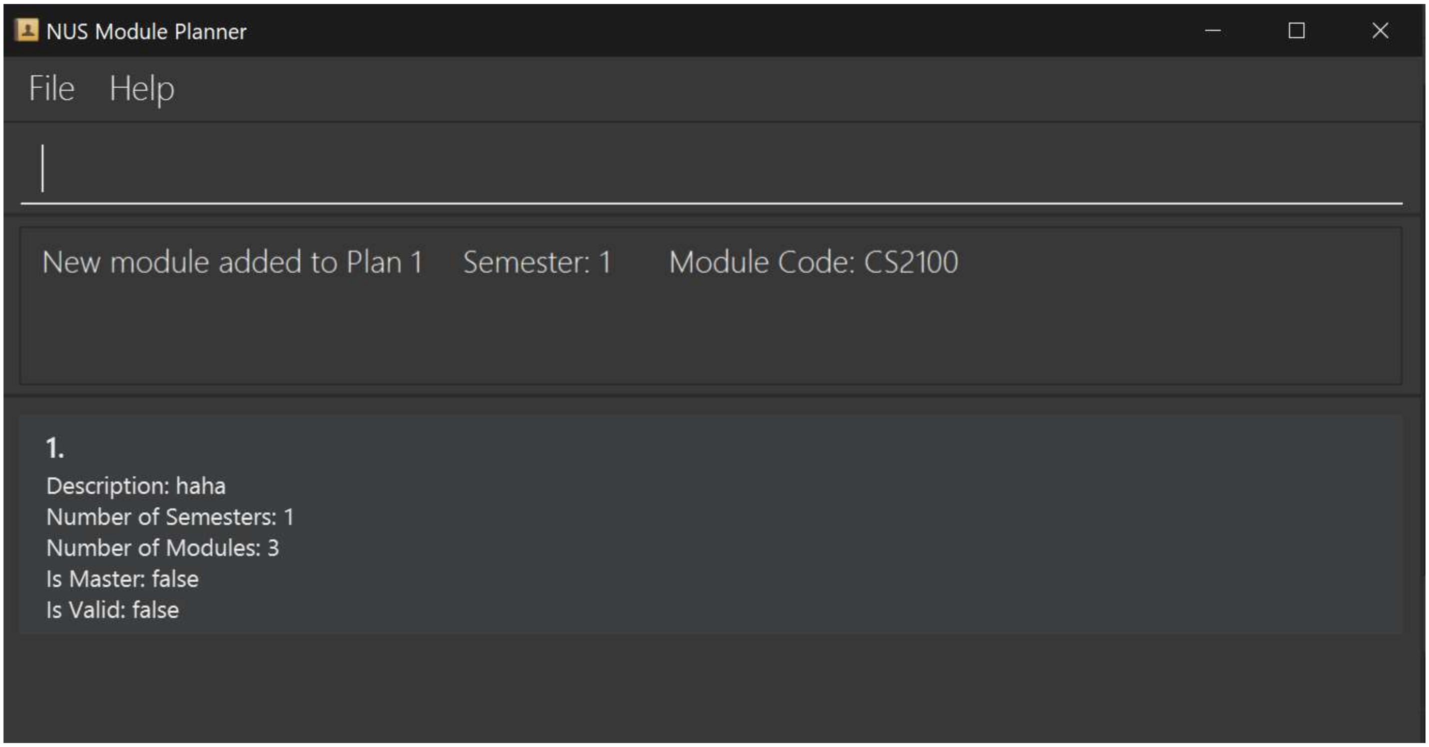
- Module addition/deletion success status
- Semester number
- Module code

Constraints:

- Trying to add a module that already exists will not be allowed
- Trying to add/delete a nonexistent module code/plan number/semester number will not be allowed

> *Note: As this is not a fully fledged production level project, there are limited modules for users to add. If in the future this project grows to a production piece, it will use relevant NUS APIs to collect modules for users to add. Instead, currently there are only dozens of modules for users to add, but for the purposes of testing and showcasing this product, it is sufficient.*

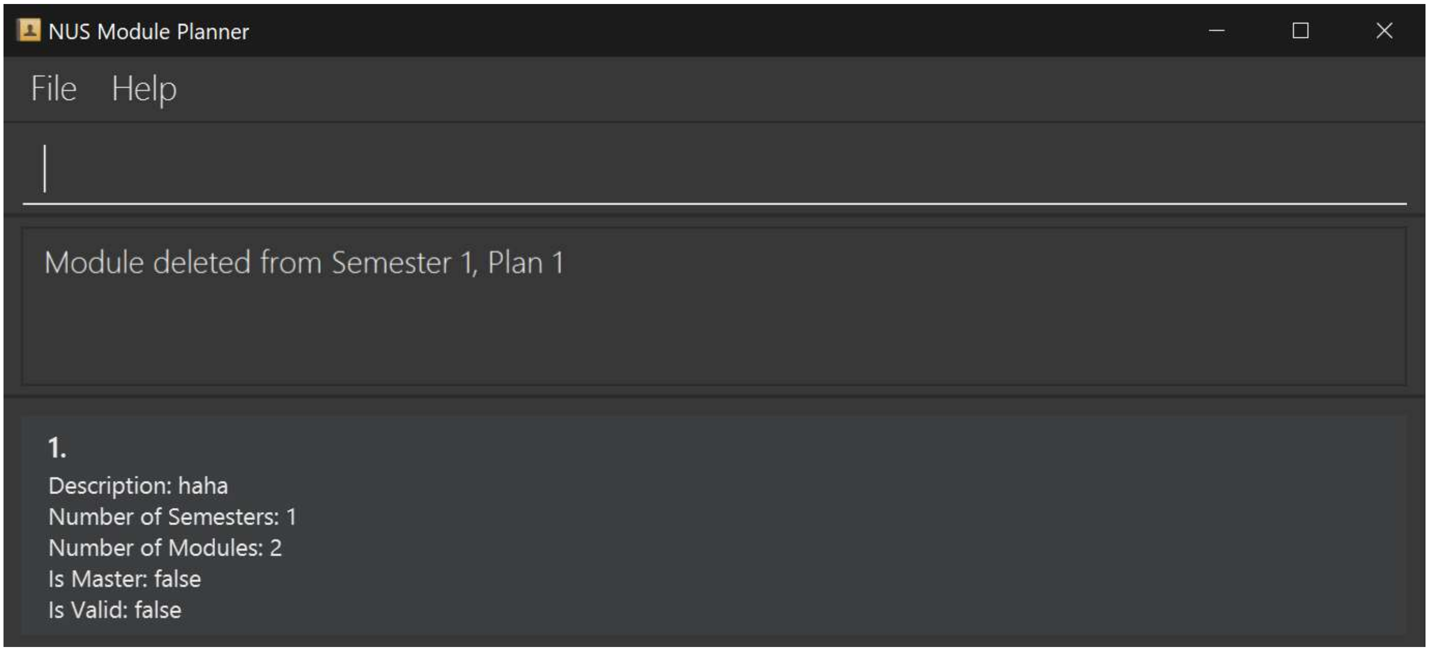
Example output for add module command:



New module added to Plan 1 Semester: 1 Module Code: CS2100

1.
Description: haha
Number of Semesters: 1
Number of Modules: 3
Is Master: false
Is Valid: false

Example output for deleting a module:



Module deleted from Semester 1, Plan 1

1.
Description: haha
Number of Semesters: 1
Number of Modules: 2
Is Master: false
Is Valid: false

View module info: `info`

Format: `info m/MODULE_CODE`

> *Tip: A user can also add a module to a plan/semester (See `add/delete`)*

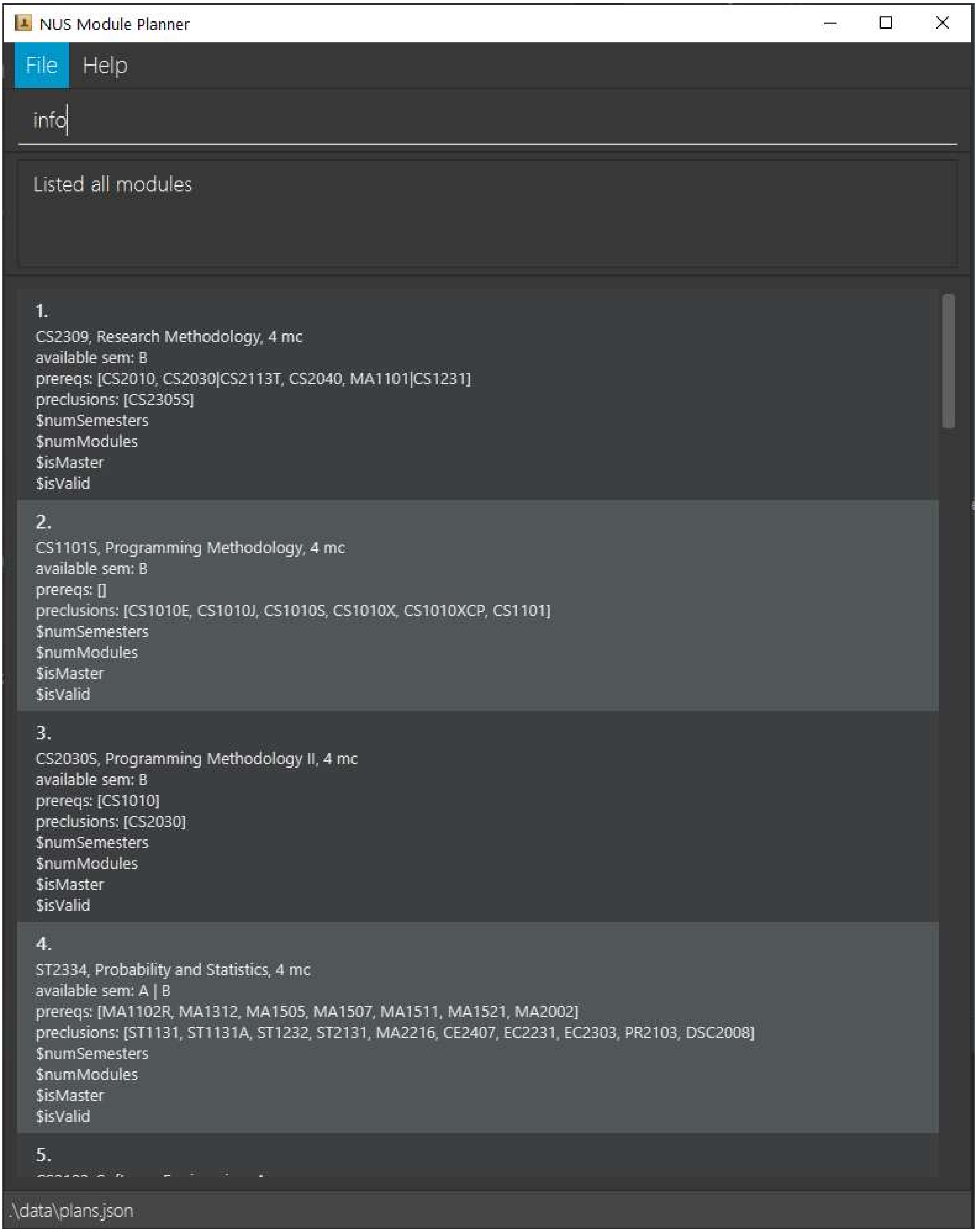
By default, this command takes in one optional argument, `MODULE_CODE` and outputs the module information including:

- Brief Description
- Number of MCs
- Semesters available
- Pre-requisites
- Preclusions

Constraints:

- Module has to exist

Example output listing all module information:



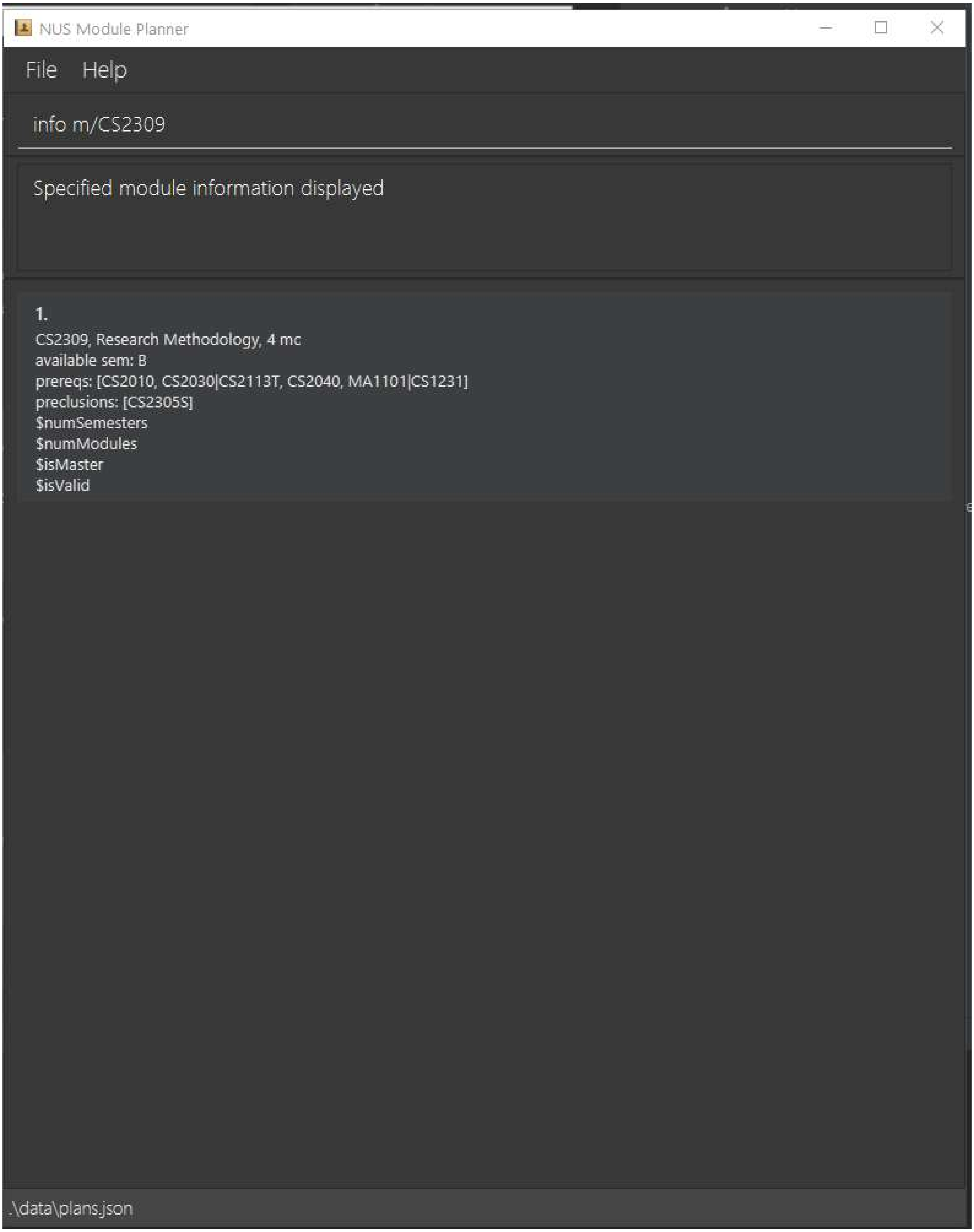
```
info
```

```
Listed all modules
```

```
1.
CS2309, Research Methodology, 4 mc
available sem: B
prereqs: [CS2010, CS2030|CS2113T, CS2040, MA1101|CS1231]
preclusions: [CS2305S]
$numSemesters
$numModules
$isMaster
$isValid
2.
CS1101S, Programming Methodology, 4 mc
available sem: B
prereqs: []
preclusions: [CS1010E, CS1010J, CS1010S, CS1010X, CS1010XCP, CS1101]
$numSemesters
$numModules
$isMaster
$isValid
3.
CS2030S, Programming Methodology II, 4 mc
available sem: B
prereqs: [CS1010]
preclusions: [CS2030]
$numSemesters
$numModules
$isMaster
$isValid
4.
ST2334, Probability and Statistics, 4 mc
available sem: A | B
prereqs: [MA1102R, MA1312, MA1505, MA1507, MA1511, MA1521, MA2002]
preclusions: [ST1131, ST1131A, ST1232, ST2131, MA2216, CE2407, EC2231, EC2303, PR2103, DSC2008]
$numSemesters
$numModules
$isMaster
$isValid
5.
```

```
.\data\plans.json
```

Example output for finding a specific module information



info m/CS2309

Specified module information displayed

1.
CS2309, Research Methodology, 4 mc
available sem: B
prereqs: [CS2010, CS2030|CS2113T, CS2040, MA1101|CS1231]
preclusions: [CS2305S]
$numSemesters
$numModules
$isMaster
$isValid

.\data\plans.json

Overall input constraints

The NUS Module Planner app is designed to help students track their typical study plans. Hence, user inputs for indexes such as plan numbers or semester numbers are only guaranteed to work within reasonable limits. For example a student should never have 2 billion semesters in their study plan as this would exceed the typical human lifetime.

Explicitly, the following should take on values between 1 and 100 only. Correct behaviour of the app is not guaranteed beyond these bounds.

- Semester number
- Plan number
- Number of modules in a plan or semester

Please note that if you manually edit the `moduleinfo.json` or the `plans.json` files that the NUS Module Planner application may not longer function as expected.