CodeCampX - User Guide

By: `Team CS2103-AY1920S2-CS2103-W14` Since: `Jan 2020` Licence: `MIT`

1. Welcome to CodeCampX

Code Camp X (CCX) is for those who **prefer to use a desktop app to manage the administrative tasks of running a coding camp**. More importantly, CCX is **optimized for those who prefer to work with a Command Line Interface** (CLI) while still having the benefits of a Graphical User Interface (GUI). If you can type fast, CCX can allow you to manage your camp's administrative tasks faster than traditional GUI apps. Interested? Jump to the Section 2, "Quick Start" to get started. Enjoy!

1.1. Legend

The following **3** callouts will be used throughout the documentation which you may wish to pay attention to as it may contain important details:

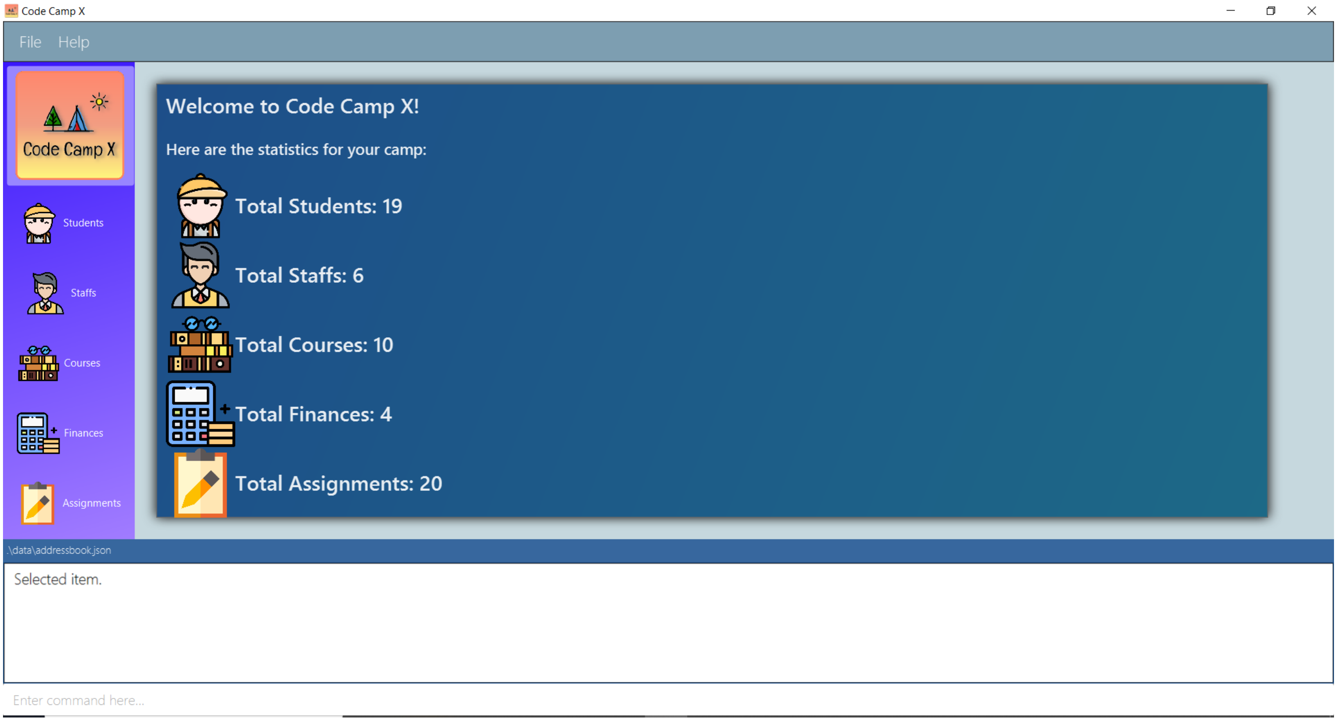
NOTE Just for your info, do not be alarmed. Be sure to read these notes as it might contain some important information.

TIP Perhaps something can be done using another approach, but it is up to you to decide. Tips are often not important and can be safely ignored.

WARNING Some things might go wrong if you are not careful, or did not follow the instructions correctly. You are strongly advised to read whatever is in this block.

1.2. A Quick Look at CodeCampX

CodeCampX's user interface is split into a few sections. These sections are highlighted in the image below.



Welcome to Code Camp X!

Here are the statistics for your camp:

Total Students: 19

Total Staffs: 6

Total Courses: 10

Total Finances: 4

Total Assignments: 20

\data\addressbook.json

Selected item.

Enter command here...

Figure 1. Quick Overview of CodeCampX.

Let's get started!

2. Quick Start

Follow this installation guide to get **CodeCampX** up and running on your computer.

1. Ensure you have Java version `9` or later installed in your Computer.

NOTE

- If you are unsure which Java version is installed, you may refer to this link.
- You may install the current version of Java here.

1. Download the latest version here.
2. Copy the file to the folder you want to use as the home folder.
3. Double-click the file to start the app. The GUI should appear in a few seconds.
4. 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.
5. Some example commands you can try:
   - `help` : Opens up the help page
   - `exit` : Exits the application
6. Refer to Section 5, "Commands" for details of each command.

3. Components

CodeCampX consists of five core components: Students, Staffs, Courses, Finances, Assignments

3.1. Students

You can manage the students by assigning them to courses, and ensuring that they have paid for their courses.

3.2. Staffs

You can manage the staffs by providing information such as the phone number and email address. This is critical as staffs are the main point of contact during an emergency. You can keep track of which teacher staffs you have paid for teaching a course by checking via Finances.

3.3. Courses

You can keep track of the ongoing courses in your Coding Camp. Each course will have an assigned teacher, as well as a list of assigned students. The course fee should also be specified.

3.4. Finances

The built-in Sales Management component in CodeCampX provides you with the tools you will need to keep track of financial records efficiently. Several analytical features are also incorporated to assist you in financial decision-making and devising marketing strategies.

3.5. Assignments

You can keep track of the available assignments of the various students or courses through this functionality. Additionally, the progress of the assignments can be viewed for the course of the student (In Student Tab) or the student of a course (In Course Tab).

4. Key Features

4.1. Convenience

CodeCampX is an integrated application that will provide you with the utmost convenience and tools you will need to manage your Coding Camps. It allows you to:

- Export data to `.json` (default) or Excel file `[coming in v2.0]`.

4.2. Security

We understand that digital security is your biggest concern. **CodeCampX** is capable of securing your restaurants' data by:

- Encrypting all data using state of the art encryption scheme `[coming in v2.0]`.
- Providing accountability through logging of system events.

4.3. Efficiency

Time is money. **CodeCampX** ensures that the application will:

- Load within 5 seconds.
- Execute commands within split of a second and update the GUI almost instantaneously.

5. Commands

CodeCampX is jam-packed with features and it may be daunting for new users. The subsequent sections of the user guide provides a step by step walk-through of all the commands **CodeCampX** has to offer.

Do read our short explanation about Command Format below so that the subsequent portions of this section will make sense to you.

Command Format

- Words in `UPPER_CASE` are the parameters to be supplied by the user e.g. in `add-student n/STUDENT_NAME`, `STUDENT_NAME` is a parameter which can be used as `add-student n/Bob`.
- Items in square brackets are optional e.g `n/STUDENT_NAME [t/TAG]` can be used as `n/Bob t/loyal` or as `n/Bob`.
- 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/loyal`, `t/10years t/new` etc.
- Parameters can be in any order e.g. if the command specifies `cid/COURSE_ID tid/TEACHER_ID`, `tid/TEACHER_ID cid/COURSE_ID` is also acceptable.

5.1. General

The commands in this section does not tie to any of the 5 components.

5.1.1. Viewing help : `help`

Opens up the help window. Very useful if you are a new user.
Format: `help`

5.1.2. Exiting the program : `exit`

Exits the program.
Format: `exit`

5.1.3. Undo Command: `undo`

Set the app state back to the most recent undoableCommand.

NOTE See the full list of undoable commands: all edit/add/delete/assign/unassign/done/undone commands.

Examples:

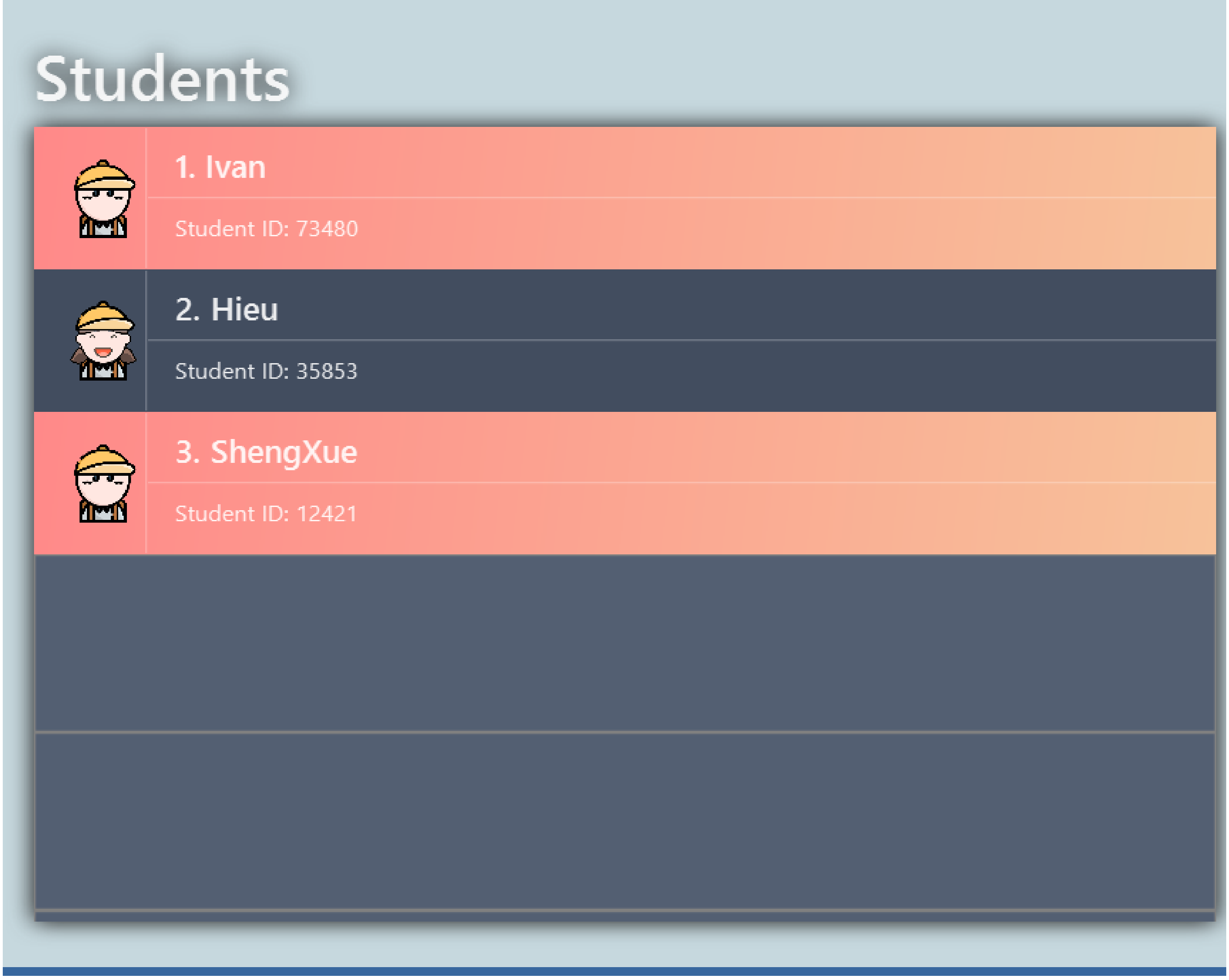
- `find-student hieu`
  `undo`

This will fail because `find-student` is not an undoableCommand.

- `delete-student 35853`
  `undo`

Undo the delete command of student and add the student back at the previous relative ordering too.

Illustration:



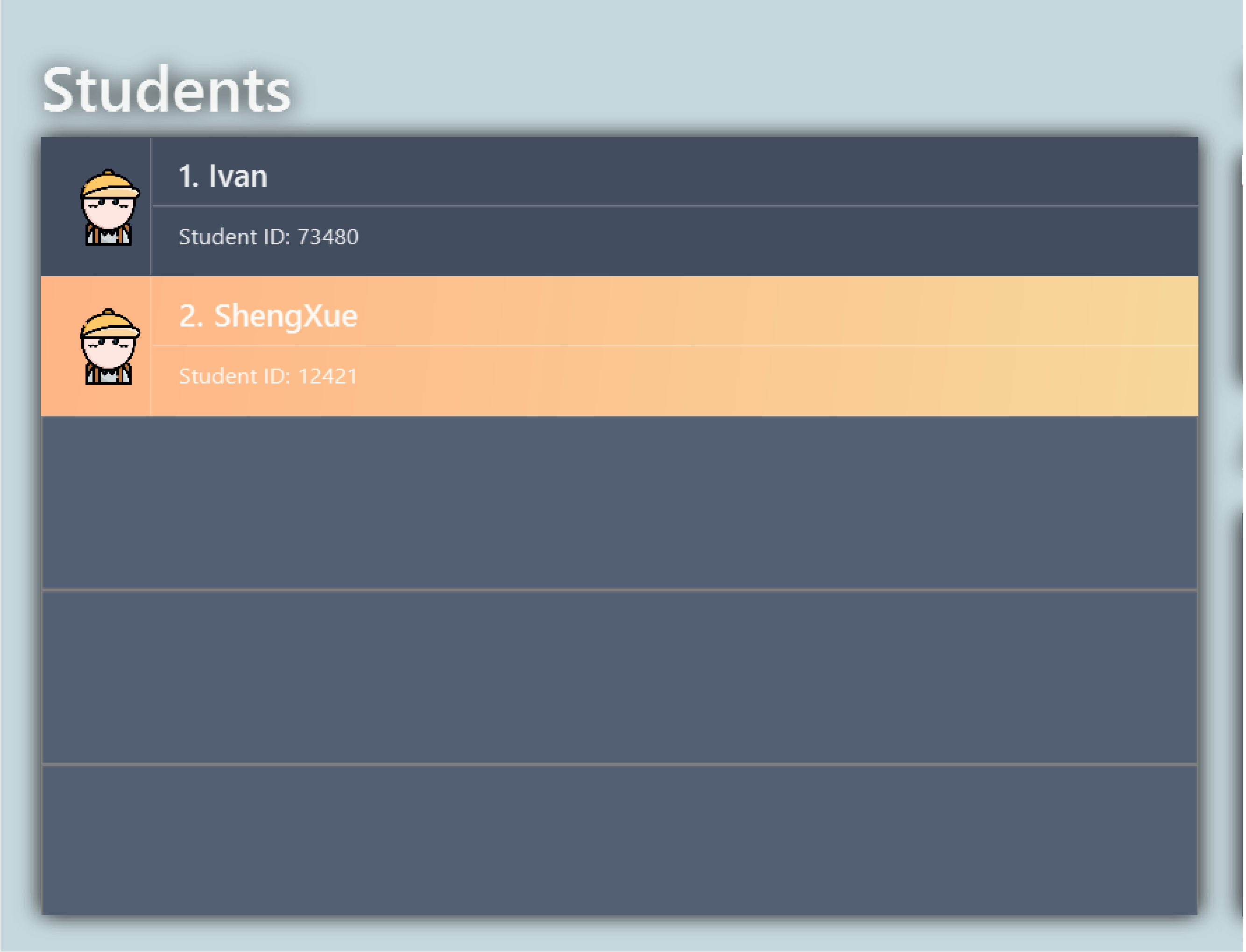
Students

1. Ivan
Student ID: 73480

2. Hieu
Student ID: 35853

3. ShengXue
Student ID: 12421

After `delete-student 35853`



Students

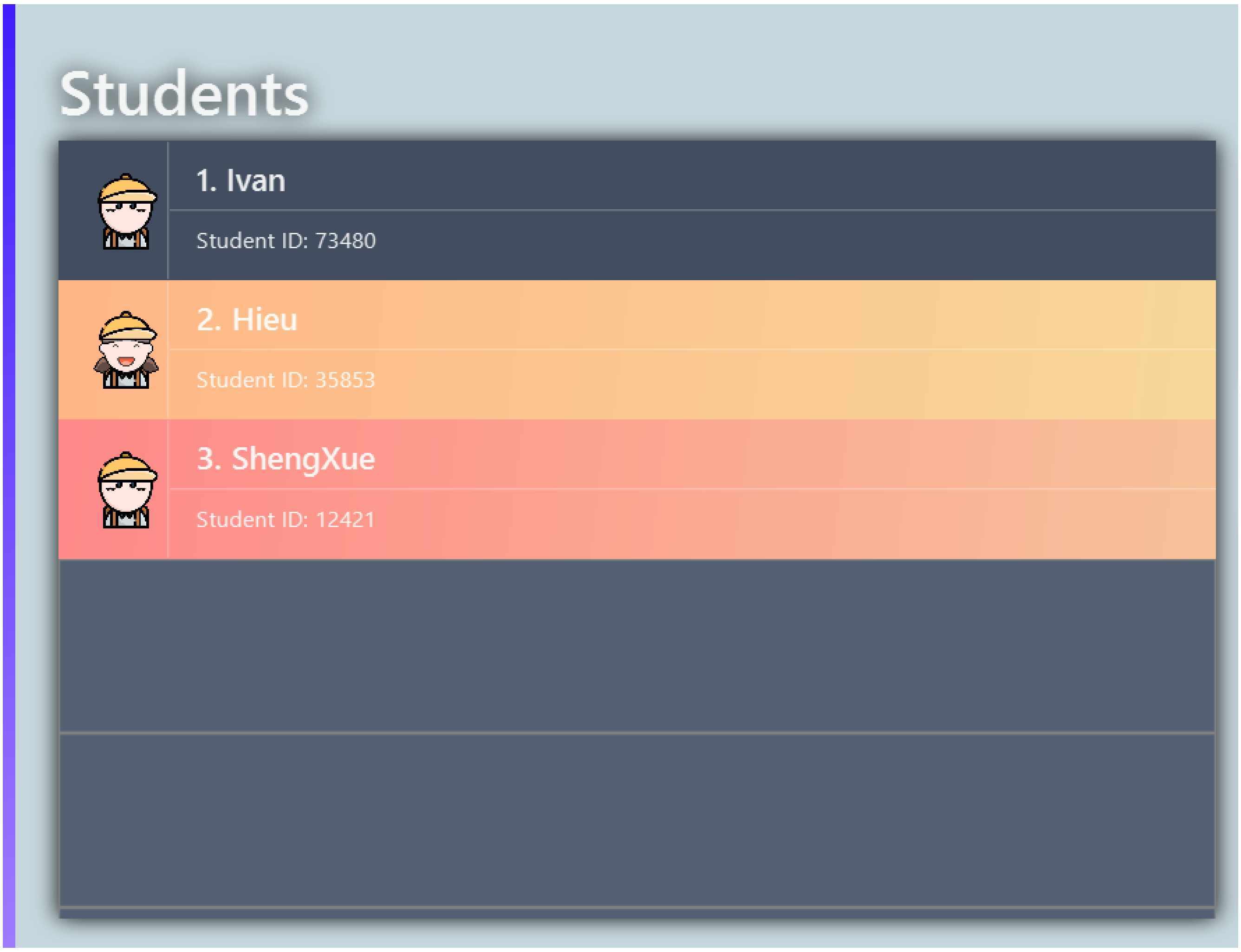
1. Ivan

Student ID: 73480

2. ShengXue

Student ID: 12421

After `undo`:



Students

1. Ivan

Student ID: 73480

2. Hieu

Student ID: 35853

3. ShengXue

Student ID: 12421

5.1.4. Redo Command: `redo`

Reverses the most recent `undo` command, but will fail if the most recent command was not the `undo` command or `redo` command

Examples:

-
  ```
  delete-student 35853
  undo
  redo
  ```

Successfully delete the student, then add the student back with `undo`, then delete student again with `redo`

-
  ```
  find-student hieu
  redo
  ```

`redo` fails because no `undo` command previously

-
  ```
  delete-student 35853
  undo
  find-student hieu
  redo
  ```

This `redo` also fails because the most recent command is not `undo` or `redo`

Format: `undo` or `redo`

5.2. Assigning/Unassigning -TYPE- to a Course : `assign` or `unassign` respectively

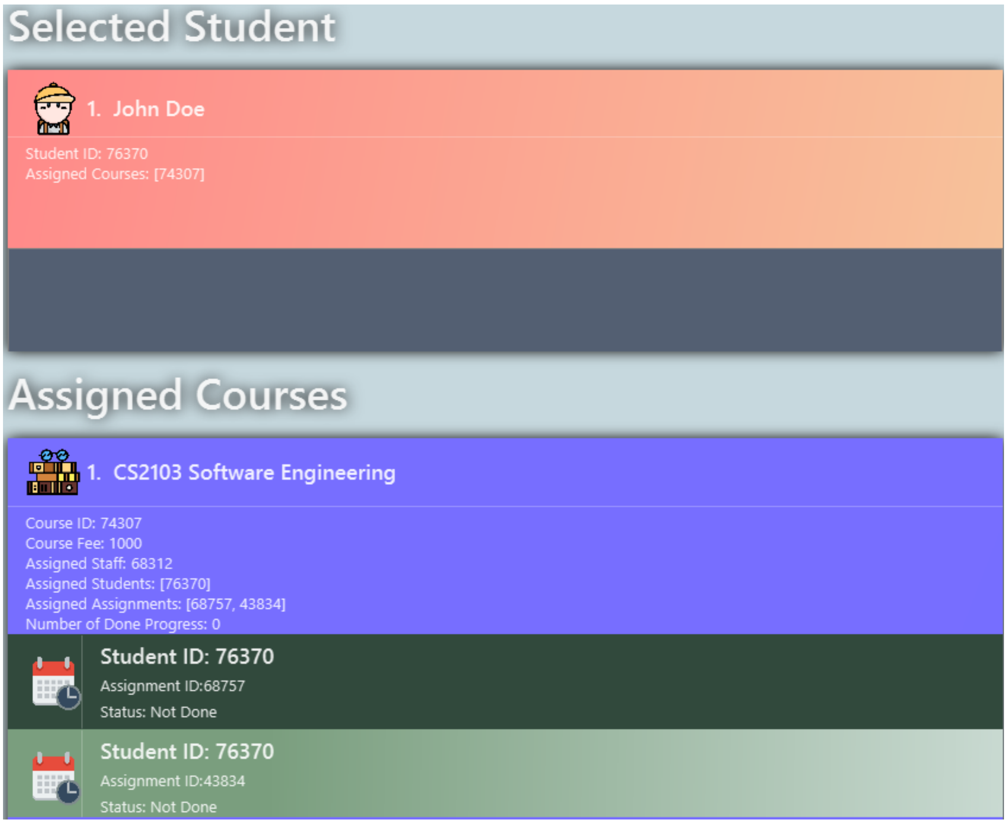
5.2.1. Assign a Student to a Course : `assign` or `unassign` respectively

Adds the Student ID to the Course ID specified.
Format: `assign cid/COURSEID sid/STUDENTID`
Example: `assign cid/829 sid/33`

Removes the Student ID from the Course ID specified.
Format: `unassign cid/COURSEID sid/STUDENTID`
Example: `unassign cid/829 sid/33`

NOTE When a student is assigned to a course, a progress will be created for the student for every assignment of the course. This progress tracks whether the student has completed the specific assignment (Done) or not (Not Done)

In Student Tab, Under Selected Student and Assigned Courses Panel, after assigning student to course, it can be seen that the progresses have been added for every assignment in the course. The Student's progresses can be tracked by specific assignment status Done/ Not done. Total done progresses can be viewed under "Number of progress done" field.



Selected Student

1. John Doe

Student ID: 76370
Assigned Courses: [74307]

Assigned Courses

1. CS2103 Software Engineering

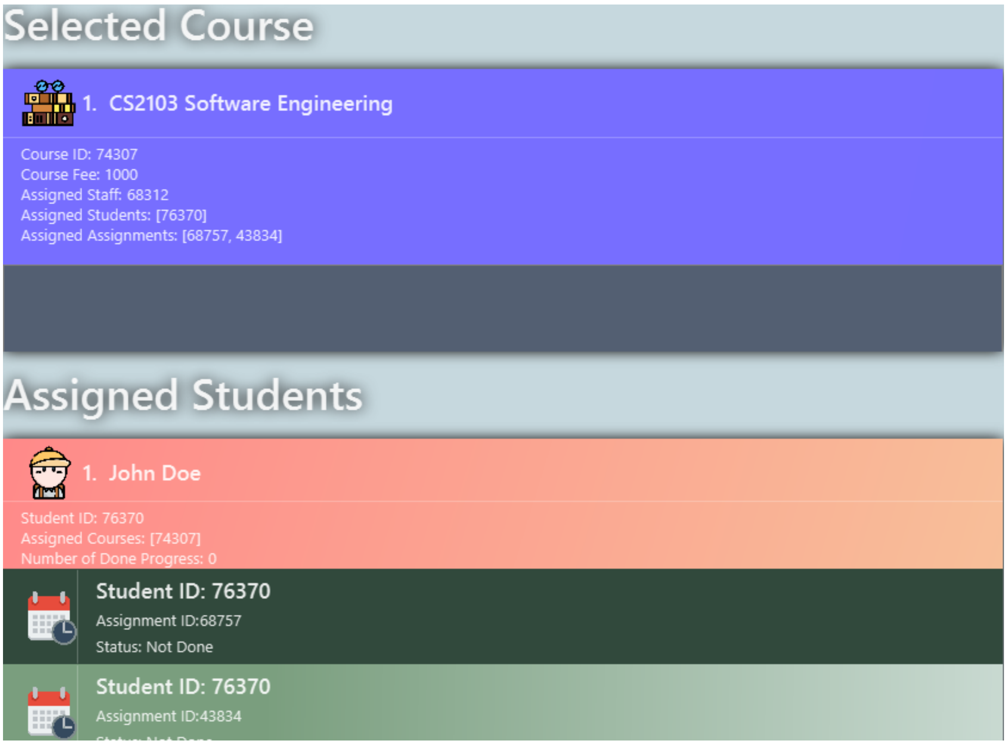
Course ID: 74307
Course Fee: 1000
Assigned Staff: 68312
Assigned Students: [76370]
Assigned Assignments: [68757, 43834]
Number of Done Progress: 0

Student ID: 76370
Assignment ID:68757
Status: Not Done

Student ID: 76370
Assignment ID:43834
Status: Not Done

Figure 2. Selected Student and Assigned Courses Panel. Shows ID and details of assigned Courses and Assignments.

Student’s details and Assigned Assignments for this courses can also be viewed under Course Tab.



Selected Course

1. CS2103 Software Engineering

Course ID: 74307
Course Fee: 1000
Assigned Staff: 68312
Assigned Students: [76370]
Assigned Assignments: [68757, 43834]

Assigned Students

1. John Doe

Student ID: 76370
Assigned Courses: [74307]
Number of Done Progress: 0

Student ID: 76370
Assignment ID:68757
Status: Not Done

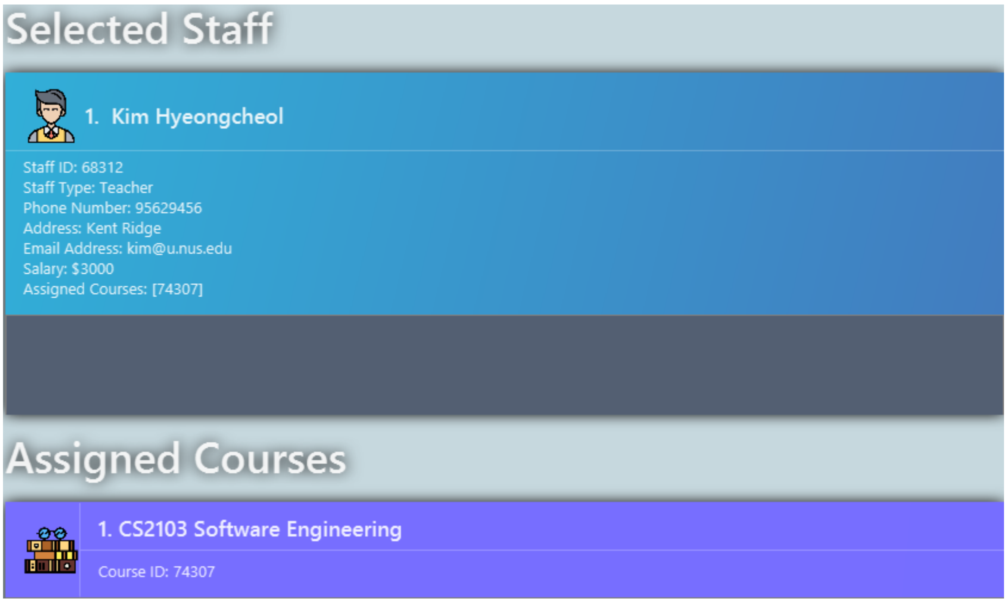
Student ID: 76370
Assignment ID:43834
Status: Not Done

Figure 3. Selected Course and Assigned Students Panel. Shows ID and details of assigned Students and their Assignments.

5.2.2. `assign`/`unassign` Teacher to/from a Course

Adds the Teacher ID to the Course ID specified.
Format: `assign cid/COURSEID tid/TEACHERID`
Example: `assign cid/829 tid/21`

When a teacher is assigned to a course, the teacher’s ID is added to course and course’s ID is added to teacher. In Selected Staff Tab, it can be seen that a course ID is added under "Assigned Courses" and the course can be viewed in Assigned Courses Tab.



Selected Staff

1. Kim Hyeongcheol

Staff ID: 68312
Staff Type: Teacher
Phone Number: 95629456
Address: Kent Ridge
Email Address: kim@u.nus.edu
Salary: $3000
Assigned Courses: [74307]

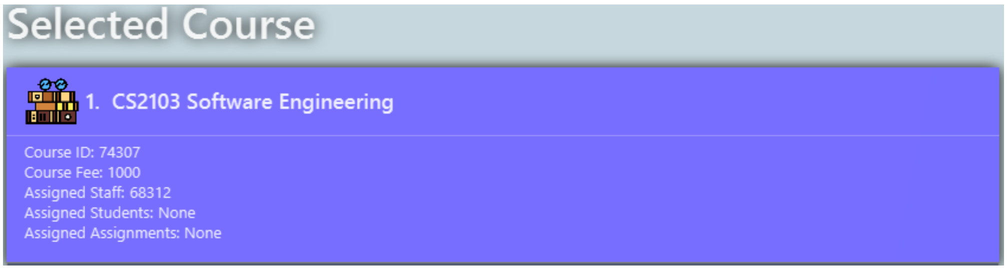
Assigned Courses

1. CS2103 Software Engineering

Course ID: 74307

Figure 4. Selected Staff and Assigned Courses Panel. Shows ID and details of assigned Courses.

In Selected Course Tab, it can be seen that a staff ID is added under "Assigned Staff".



Selected Course

1. CS2103 Software Engineering

Course ID: 74307
Course Fee: 1000
Assigned Staff: 68312
Assigned Students: None
Assigned Assignments: None

Figure 5. Selected Staff and Assigned Courses Panel. Shows ID and details of assigned Courses.

Removes the Teacher ID from the Course ID specified.
Format: `unassign cid/COURSEID tid/TEACHERID`
Example: `unassign cid/829 tid/21`

When a teacher is unassigned to a course, the teacher’s ID is removed from course and course’s ID is removed from teacher. In Selected Staff Tab, after unassign Command, the course’s ID and details no longer appears.

5.2.3. Add An Assignment to a Course : `assign` or `unassign` respectively

Adds the Assignment ID to the Course ID specified.
Format: `assign cid/COURSEID aid/ASSIGNMENTID`
Example: `assign cid/829 aid/21`

Removes the Assignment ID from the Course ID specified.
Format: `unassign cid/COURSEID aid/ASSIGNMENTID`
Example: `unassign cid/829 aid/21`

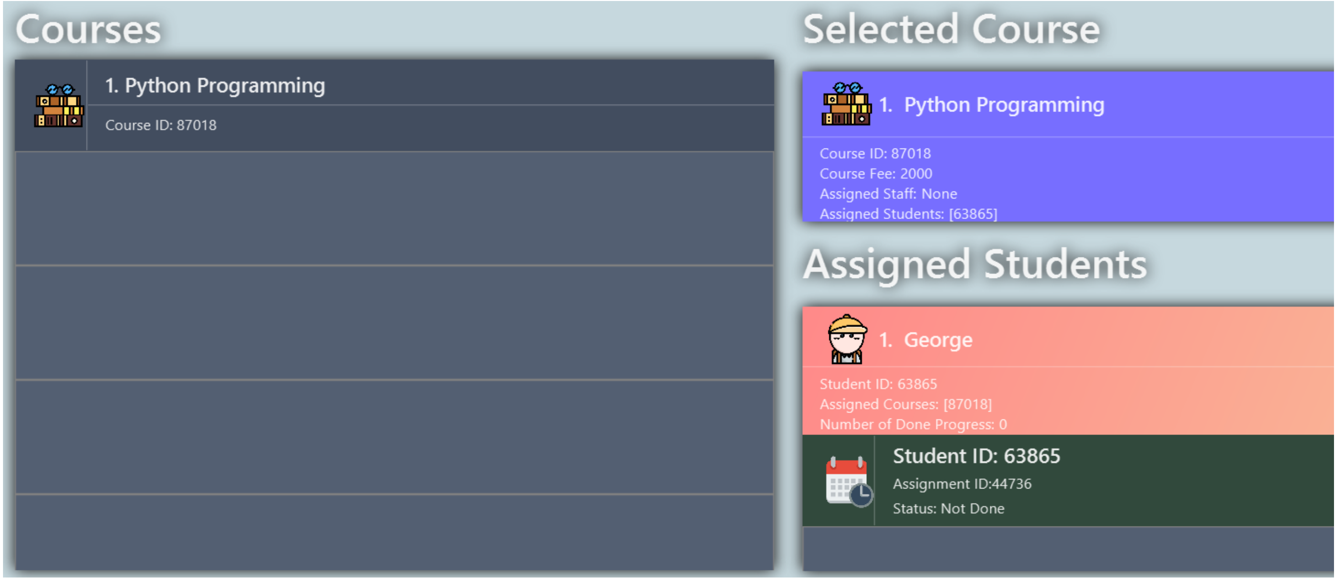
NOTE When an assignment is assigned to a course, a progress will be created for every student currently assigned to the course. This progress tracks whether the student has completed the specific assignment (Done) or not (Not Done)

5.3. Mark a Student’s Assignment as `Done`/`Undone`

Mark a student’s assignment as `Done`.
Format: `done aid/ASSIGNMENTID sid/STUDENTID`
Example: `done aid/829 sid/21`

Mark a student’s assignment as `Undone`.
Format: `undone aid/ASSIGNMENTID sid/STUDENTID`
Example: `undone aid/829 sid/21`

Illustration:



1. Python Programming
Course ID: 87018

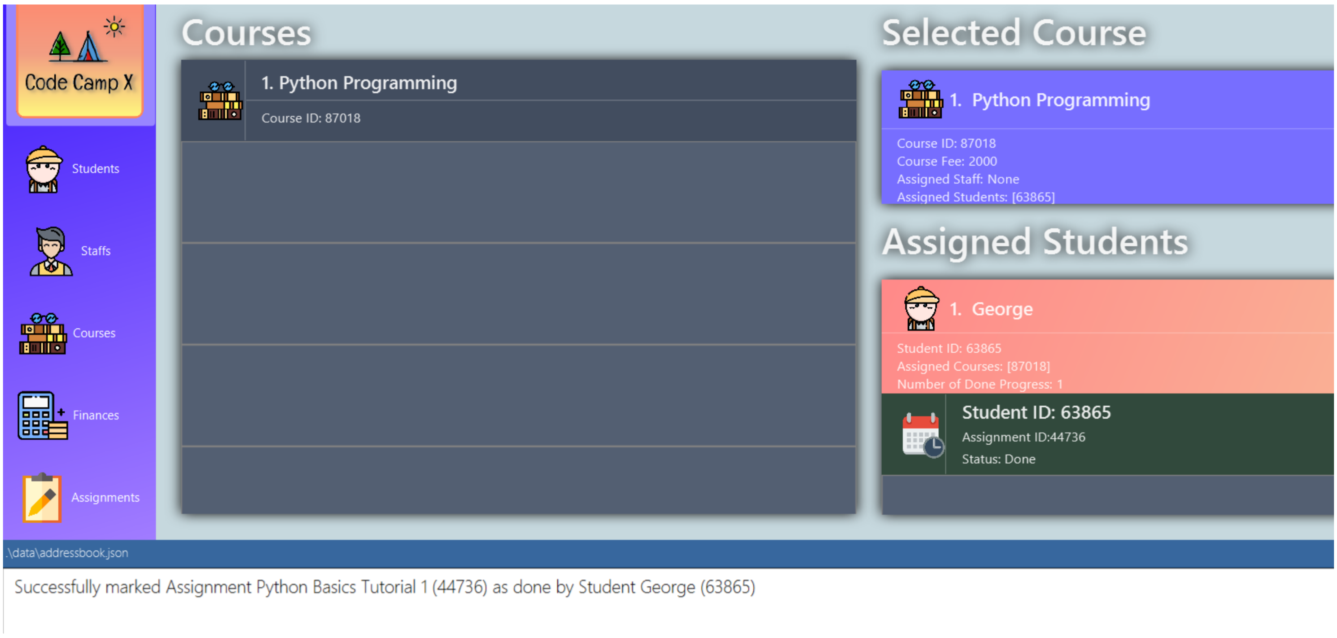
1. Python Programming
Course ID: 87018
Course Fee: 2000
Assigned Staff: None
Assigned Students: [63865]

1. George
Student ID: 63865
Assigned Courses: [87018]
Number of Done Progress: 0

Student ID: 63865
Assignment ID:44736
Status: Not Done

Notice that George has initially not completed Assignment 44736.

After `done aid/44736 sid/63865`:



1. Python Programming
Course ID: 87018

1. Python Programming
Course ID: 87018
Course Fee: 2000
Assigned Staff: None
Assigned Students: [63865]

1. George
Student ID: 63865
Assigned Courses: [87018]
Number of Done Progress: 1

Student ID: 63865
Assignment ID:44736
Status: Done

.\data\addressbook.json

Successfully marked Assignment Python Basics Tutorial 1 (44736) as done by Student George (63865)

Now observe 3 updates:

1. Success message as displayed in message box
2. The `status` of the Assignment has been marked as done
3. The `Number of Done Progress` has been incremented by 1

5.4. Select a Student/Staff/Course : `select` or simply click the bar

Selecting means to view the specified item in detail. As mentioned in the previous section, a course may be assigned student/s, a teacher staff, or assignment/s. When you select a course for example, you can view the list of students assigned in the list on the right.

If the specified ID does not exist, the selected panel will become empty.

5.4.1. Select a Student

Views details of a student and list of courses of a student
Format: `select sid/STUDENTID`
Example: `select sid/11` or click the desired Student bar

5.4.2. Select Assignment belonging to the Course of a Student

Views list of Assignment belong to Course of a Student
Format: `select sid/STUDENTID cid/COURSEID`
Example: `select sid/11 cid/1` or click the desired Course bar under the Student

5.4.3. Select a Staff

Views the details of a staff and list of courses of a staff
Format: `select tid/STAFFID`
Example: `select tid/31` or click the desired Staff bar

5.4.4. Select a Course

Views the details of a course and list of students of a course
Format: `select cid/COURSEID`
Example: `select cid/1` or click the desired Course bar

5.4.5. Select Assignment belonging to the Student of a Course:

Views list of Assignment belong to Student of a Course
Format: `select cid/COURSEID sid/STUDENTID`
Example: `select cid/1 sid/11` or click the desired Student bar under the Course

5.4.6. Select a Finance

Views the details of a finance
Format: `select fid/FINANCEID`
Example: `select fid/801` or click the desired Finance bar

5.4.7. Select an Assignment

Views the details of an assignment
Format: `select aid/ASSIGNMENTID`
Example: `select aid/901` or click the desired Assignment bar

5.5. Switch between tabs : `[CTRL] + [UP ARROW]`, `[CTRL] + [DOWN ARROW]` `m`, `s`, `t`, `c`, `f`, `a`

- The tab orders are: Summary, Staff, Course, Finance, Assignment
- Switch quickly to previous Tab (with wrap around) : Click command box + `[CTRL]` + `[UP ARROW]`
- Switch quickly to next Tab (with wrap around): Click command box + `[CTRL]` + `[DOWN ARROW]`
- Switch quickly to Summary Tab : `m` or `summary`
- Switch quickly to Student Tab : `s` or `student`
- Switch quickly to Staff Tab : `t` or `staff`
- Switch quickly to Course Tab : `c` or `course`
- Switch quickly to Finance Tab : `f` or `finance`
- Switch quickly to Assignment Tab : `a` or `assignment`

5.5.1. Navigate through your command history: `[UP ARROW]` or `[DOWN ARROW]`

- Just like a typical CLI application, you can use up and down arrow keys to navigate through your command history
- Click the command box and press `[UP ARROW]` or `[DOWN ARROW]`
- Only complete and successfully executed commands will be added to this history
- There will not be two identical commands beside each other

5.6. Saving the data

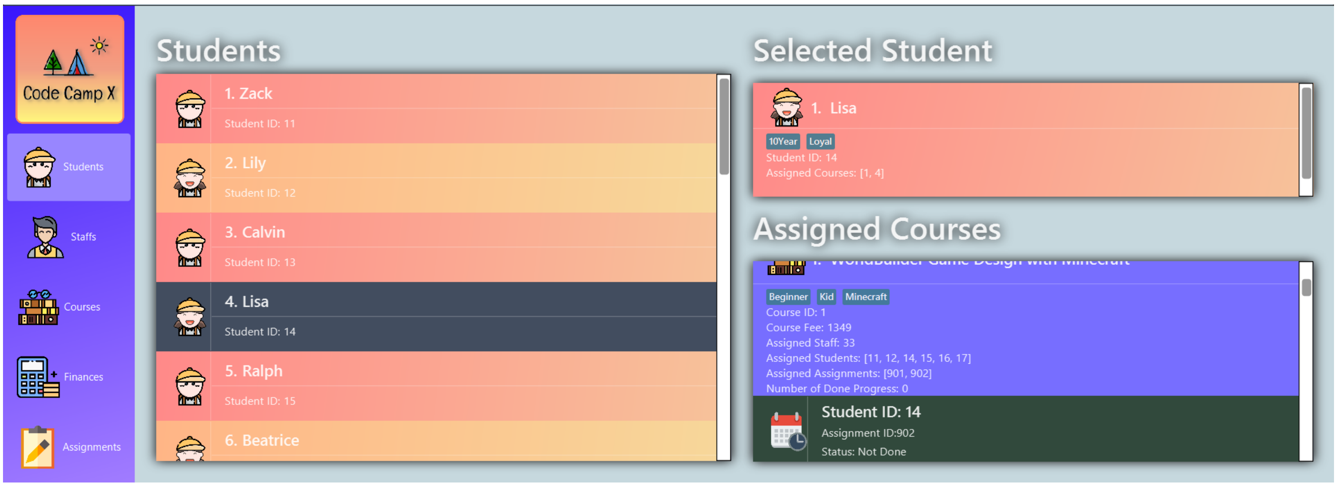
Coding Camp book data are saved in the hard disk automatically after any command that changes the data. No manual saving is required.

5.7. Clearing all the data

Coding Camp book data can be cleared by `clear-all` command. This action will **CLEAR** all the entire address book and this action can not undone.

WARNING This will irrevocably clear the entire address book and you cannot undo it! Be very careful with this command.

5.8. Student



Students

1. Zack
Student ID: 11

2. Lily
Student ID: 12

3. Calvin
Student ID: 13

4. Lisa
Student ID: 14

5. Ralph
Student ID: 15

6. Beatrice

Selected Student

1. Lisa
10Year Loyal
Student ID: 14
Assigned Courses: [1, 4]

Assigned Courses

1. WorldBuilder Game Design with Minecraft
Beginner Kid Minecraft
Course ID: 1
Course Fee: 1349
Assigned Staff: 33
Assigned Students: [11, 12, 14, 15, 16, 17]
Assigned Assignments: [901, 902]
Number of Done Progress: 0

Student ID: 14
Assignment ID:902
Status: Not Done

Figure 6. List of Students. Shows details and list of courses of selected student. The list of Assignments is also shown for the selected course

5.8.1. Add a student: `add-student`

Add a new student to the list of student
Format: `add student n/NAME g/GENDER [t/TAG]…`
Example: * `add student n/Jon Snow g/m t/Hardworking`

TIP	The Gender of the Student is shown in the form of a little icon on the left. This automatically updates when you `edit-student 12 g/f` or `edit-student 12 g/m` as well. Try it!

5.8.2. Remove a student: `delete-student`

Remove a student from the list of students
Format: `delete-student ID`
Example:

- `delete-student 16100`

> - Deletes the item at the specified `ID`. The ID refers to the ID number shown in the displayed item panel list
> - The ID **must be a positive integer** 16100, 25200, 55250, …

NOTE	Be reminded that when you delete a student, the student will be deleted from every course as well For example, a course that was only assigned to this student **Bob** will change to **None** if student **Bob** is deleted.

5.8.3. Locating items by name: `find-student`

Finds student whose names contain any of the given keywords.
Format: `find-student KEYWORD [MORE_KEYWORDS]…`

- The search is case insensitive. e.g `bob` will match `Bob`
- The order of the keywords does not matter. e.g. `Ned Stark` will match `Stark Ned`
- Only the name is searched
- Only full words will be matched e.g. `Sta` will not match `Stark`
- Items matching at least one keyword will be returned (i.e. `OR` search). e.g. `Ned Stark` will return `Ned Stark`, `Bran Stark` and `Ned Targaryen`

Examples:

- `find-student Ned`
  Returns `ned` and `Ned Stark`
- `find-student Lannister Targaryen Stark`
  Returns any student having names `Lannister`, `Targaryen`, or `Stark`

5.8.4. List all students: `list-student`

Shows a list of all students Format: `list-student`

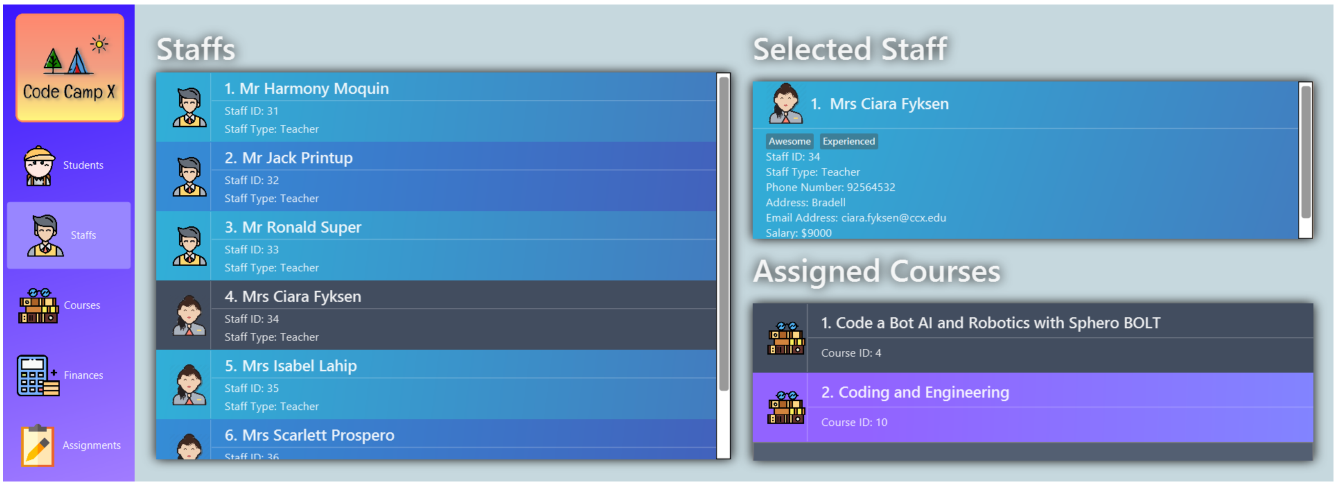
5.8.5. Edit information about a student: `edit-student`

Edits an existing student in the student list Format: `edit-student ID [n/NAME] [g/GENDER] [t/TAG]…`
Example:

- `edit-student 16100 n/Aegon Targaryen t/freshman`

- Edits the student at the specified `ID`. The ID refers to the ID number shown in the displayed student panel list
- The ID **must be a positive integer** 16100, 25200, 52500, …
- At least one of the optional fields must be provided
- Existing values will be updated to the input values
- When editing tags, the existing tags of the item will be removed i.e adding of tags is not cumulative
- You can remove all the student’s tags by typing `t/` without specifying any tags after it

5.9. Staff



Staffs

1. Mr Harmony Moquin
Staff ID: 31
Staff Type: Teacher

2. Mr Jack Printup
Staff ID: 32
Staff Type: Teacher

3. Mr Ronald Super
Staff ID: 33
Staff Type: Teacher

4. Mrs Ciara Fyksen
Staff ID: 34
Staff Type: Teacher

5. Mrs Isabel Lahip
Staff ID: 35
Staff Type: Teacher

6. Mrs Scarlett Prospero
Staff ID: 36

Selected Staff

1. Mrs Ciara Fyksen
Awesome Experienced
Staff ID: 34
Staff Type: Teacher
Phone Number: 92564532
Address: Bradell
Email Address: ciara.fyksen@ccx.edu
Salary: $9000

Assigned Courses

1. Code a Bot AI and Robotics with Sphero BOLT
Course ID: 4

2. Coding and Engineering
Course ID: 10

Figure 7. List of Staffs. Shows details and list of courses of selected staff

5.9.1. Add a staff: `add-staff`

Add a new staff to the list of staff
Format: `add-staff n/NAME lvl/LEVEL g/GENDER p/PHONE e/EMAIL s/SALARY a/ADDRESS [t/TAG]…`

Example: * `add-staff n/Bob Ross lvl/teacher g/m p/98765432 e/bob.ross@gmail.com s/1000 a/311, Clementi Ave 2, #02-25 t/LovesArt t/Friendly`

WARNING

The lvl of the staff refers to the Staff Type. It can only be specified as `teacher` or `admin`. Only a `teacher` can be assigned to a course. An `admin` cannot be assigned to a course.
A `teacher` is paid by the courses that he/she teaches. (finance type ft/ct) An `admin` is paid by miscellaneous transactions. (finance type ft/m)

TIP

The Gender of the Staff is shown in the form of a little icon on the left. This automatically updates when you `edit-staff 31 g/f` or `edit-staff 31 g/m` as well. Try it!

5.9.2. Remove a staff: `delete-staff`

Remove a staff from the list of staffs
Format: `delete-staff ID`
Example:

- `delete-staff 16100`

> - Deletes the item at the specified `ID`. The ID refers to the ID number shown in the displayed item panel
> - The ID **must be a positive integer** 16100, 25200, 52500, …

NOTE

Be reminded that when you delete a staff that is of staff type: Teacher, the teacher will be deleted from every course as well.
For example, a course that was assigned teacher **Bob** will change to **None** if teacher **Bob** is deleted.

5.9.3. Locating items by name: `find-staff`

Finds staff whose names contain any of the given keywords.
Format: `find-staff KEYWORD [MORE_KEYWORDS]…`

- The search is case insensitive. e.g `bob` will match `Bob`
- The order of the keywords does not matter. e.g. `Ned Stark` will match `Stark Ned`
- Only the name is searched
- Only full words will be matched e.g. `Sta` will not match `Stark`
- Items matching at least one keyword will be returned (i.e. `OR` search). e.g. `Ned Stark` will return `Ned Stark`, `Bran Stark` and `Ned Targaryen`

Examples:

- `find-staff Ned`
  Returns `ned` and `Ned Stark`
- `find-staff Lannister Targaryen Stark`
  Returns any staffs having names `Lannister`, `Targaryen`, or `Stark`

5.9.4. List all staffs: `list-staff`

Shows a list of all staffs Format: `list-staff`

5.9.5. Edit information about a staff: `edit-staff`

Edits an existing staff in the staff list Format: `edit-staff ID [n/NAME] [g/GENDER] [p/PHONE] [e/EMAIL] [s/SALARY] [a/ADDRESS] [t/TAG]…`
Example:

- `edit-staff 16100 n/Aegon Targaryen t/legend`

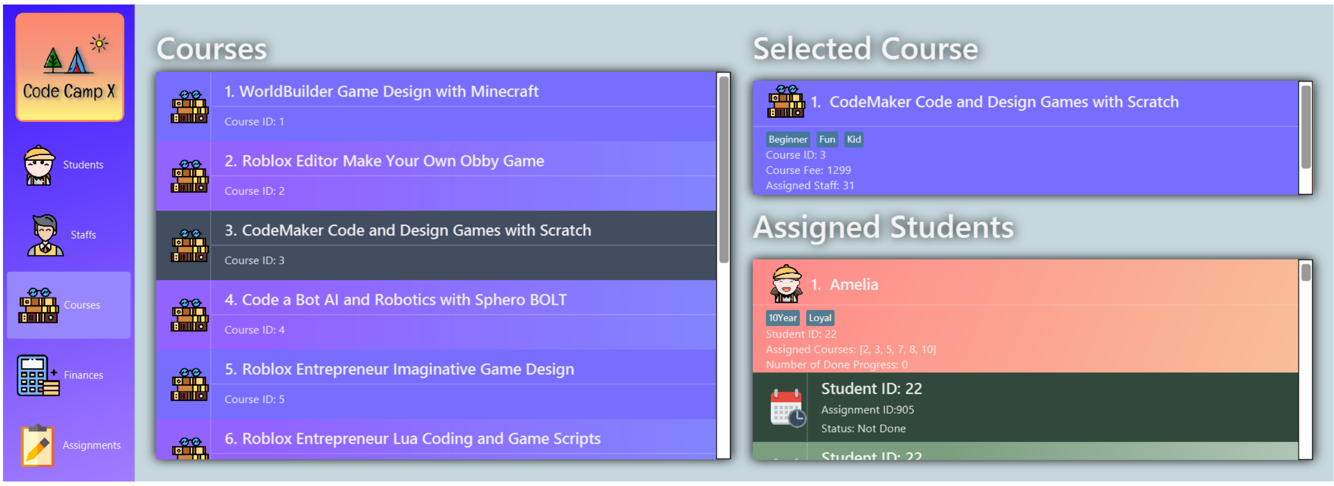
- Edits the staff at the specified `ID`. The ID refers to the ID number shown in the displayed staff panel
- The ID **must be a positive integer** 16100, 25200, 52500, …
- At least one of the optional fields must be provided
- Existing values will be updated to the input values
- When editing tags, the existing tags of the item will be removed i.e adding of tags is not cumulative
- You can remove all the staff’s tags by typing `t/` without specifying any tags after it

NOTE

You only can edit Name, Gender, Phone, Email, Salary, Address and Tags of a Staff.

If you edit a salary of a Teacher Staff, all the Finance associated with him still **remain the same**. This is because a Finance is added based on the salary of a Teacher at that moment and it records the transaction made. If a Teacher Staff’s salary is edited (for example: increased), it means this Teacher’s salary base is increased and it has nothing to do with the Finances that has been made.

5.10. Course



1. WorldBuilder Game Design with Minecraft
Course ID: 1

2. Roblox Editor Make Your Own Obby Game
Course ID: 2

3. CodeMaker Code and Design Games with Scratch
Course ID: 3

4. Code a Bot AI and Robotics with Sphero BOLT
Course ID: 4

5. Roblox Entrepreneur Imaginative Game Design
Course ID: 5

6. Roblox Entrepreneur Lua Coding and Game Scripts

1. CodeMaker Code and Design Games with Scratch
Beginner Fun Kid
Course ID: 3
Course Fee: 1299
Assigned Staff: 31

1. Amelia
10Year Loyal
Student ID: 22
Assigned Courses: [2, 3, 5, 7, 8, 10]
Number of Done Progress: 0

Student ID: 22
Assignment ID:905
Status: Not Done

Student ID: 22

Figure 8. List of Courses. Shows details and list of students of selected Course. The list of Assignments is also shown for the selected Student

5.10.1. Add a course: `add-course`

Add a new course to the list of course
Format: `add course n/NAME a/AMOUNT [t/TAG]…`
Example: `add course n/Cozmo Programming a/2000 t/Fun t/Robot`

5.10.2. Remove a course: `delete-course`

Remove a course from the list of courses
Format: `delete-course ID`
Example:

- `delete-course 16100`

NOTE

Be reminded that when you delete a course, the course will be deleted from every student and teacher as well.
For example, a teacher that was only assigned to this course **Java** will change to **None** if course **Java** is deleted.

5.10.3. Edit information about a course: `edit-course`

Edits an existing course in the course list Format: `edit-course ID [n/NAME] [a/AMOUNT] [t/TAG]…`
Example:

- `edit-course 1 n/Java 101 a/1000 t/intermediate`

> - Edits the course at the specified `ID`. The ID refers to the ID number shown in the displayed course panel
> - The ID **must be a positive integer** 16100, 25200, 52500, …
> - At least one of the optional fields must be provided
> - Existing values will be updated to the input values
> - When editing tags, the existing tags of the item will be removed i.e adding of tags is not cumulative
> - You can remove all the course's tags by typing `t/` without specifying any tags after it

5.10.4. Locating items by name: `find-course`

Finds course whose names contain any of the given keywords.
Format: `find-course KEYWORD [MORE_KEYWORDS]…`

> - The search is case insensitive. e.g `Cozmo` will match `cozmo`
> - The order of the keywords does not matter. e.g. `Cozmo Programming` will match `Programming Cozmo`
> - Only the name is searched
> - Only full words will be matched e.g. `Coz` will not match `Cozmo`
> - Items matching at least one keyword will be returned (i.e. `OR` search). e.g. `Cozmo Programming` will return `Cozmo Programming`, `Cozmo Introduction` and `Java Programming`

Examples:

- `find-course Cozmo`
  Returns `cozmo` and `Cozmo Programming`
- `find-course Java Python C`
  Returns any course having names `Java`, `Python`, or `C`

5.10.5. List all courses: `list-course`

Shows a list of all courses Format: `list-course`

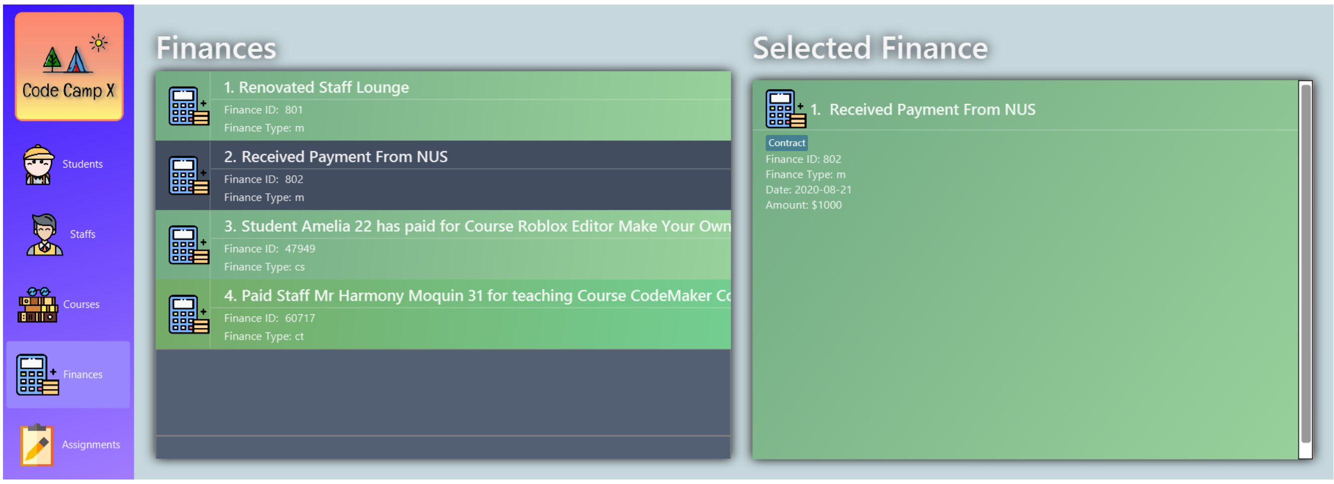
5.10.6. Edit information about a course: `edit-course`

Edits an existing course in the course list Format: `edit-course ID [n/NAME] [a/AMOUNT] [t/TAG]…` Example:

- `edit-course 16100 n/Java Programming a/2000`

> - Edits the course at the specified `ID`. The ID refers to the ID number shown in the displayed course panel
> - The ID **must be a positive integer** 16100, 25200, 55250, …
> - At least one of the optional fields must be provided
> - Existing values will be updated to the input values
> - When editing tags, the existing tags of the item will be removed i.e adding of tags is not cumulative
> - You can remove all the course’s tags by typing `t/` without specifying any tags after it

5.11. Finance



1. Renovated Staff Lounge
Finance ID: 801
Finance Type: m

2. Received Payment From NUS
Finance ID: 802
Finance Type: m

3. Student Amelia 22 has paid for Course Roblox Editor Make Your Own
Finance ID: 47949
Finance Type: cs

4. Paid Staff Mr Harmony Moquin 31 for teaching Course CodeMaker Cc
Finance ID: 60717
Finance Type: ct

1. Received Payment From NUS
Finance ID: 802
Finance Type: m
Date: 2020-08-21
Amount: $1000

Figure 9. List of Finances. Shows details of selected Finance. Type m is Finance Type 1, Type cs is Finance Type 2, and Type ct is Finance Type 3

5.11.1. Add a finance: `add-finance`

Add a new finance to the list of finance

- Type 1: Adding miscellaneous transactions(Specify ft/ as m)
  - Parameters: ft/FINANCETYPE d/DATE n/NAME a/AMOUNT [t/TAG]…
  - Example: add-finance ft/m d/2020-12-09 n/Paid NTU a/1200 t/Partnership t/Monthly
- Type 2: A student paying for a course(Specify ft/ as cs)
  - Parameters: ft/FINANCETYPE d/DATE cid/COURSEID sid/STUDENTID [t/TAG]…
  - Example: add-finance ft/cs d/2020-12-09 cid/829 sid/33 t/Late
- Type 3: A teacher is paid for teaching a course(Specify ft/ as ct)
  - Parameters: ft/FINANCETYPE d/DATE cid/COURSEID tid/TEACHERID [t/TAG]…
  - Example: add-finance ft/ct d/2020-12-09 cid/829 tid/21 t/Early

To summarise, all three types requires ft/FINANCETYPE and d/DATE to be specified.
Type 1 additionally requires n/FINANCE-NAME a/AMOUNT
Type 2 additionally requires cid/COURSEID sid/STUDENTID
Type 3 additionally requires cid/COURSEID tid/TEACHERID

TIP

For Type 1 miscellaneous transactions, you can specify the amount as negative(with a "-" in front) or positive(without a "-" in front) For example, a/100 or a/-100

For Type 2 student paying for a course, the amount will be positive automatically, as it is an earning

For Type 3 teacher being paid for teaching a course, the amount will be negative automatically, as it is an expense

To make Payment for a teacher, it is important to check the salary value of the teacher. This value is set to 0 by default when adding a new staff (teacher) if the user does not specify salary. A result of payment with amount 0 will be added if the salary of the teacher is 0.

The positive and negative amount is important (for miscellaneous type) as it is used under "earnings" and "expenses" commands.

5.11.2. Remove a finance: `delete-finance`

Remove a finance from the list of finances
Format: `delete-finance ID`
Example:

- `delete-finance 16100`

- Deletes the item at the specified `ID`. The ID refers to the ID number shown in the displayed item panel. Each finance has a unique ID number
- The ID **must be a positive integer** 16100, 25270, …

5.11.3. Edit a finance

Edits an existing finance in the finance list Format: `edit-finance ID [n/NAME] [d/DATE] [a/AMOUNT] [t/TAG]…`
Example:

- `edit-finance 16100 n/Buying new tables d/2020-04-13 a/3000`

> - Edits the finance with the specified `ID`. The ID refers to the ID number shown in the displayed payment panel
> - The ID **must be a positive integer** 16100, 25200, 55250, …
> - At least one of the optional fields must be provided
> - Existing values will be updated to the input values
> - When editing tags, the existing tags of the item will be removed i.e adding of tags is not cumulative
> - You can remove all the payment's tags by typing `t/` without specifying any tags after it

5.11.4. Locating items by name: `find-finance`

Finds finance whose names contain any of the given keywords.
Format: `find-finance KEYWORD [MORE_KEYWORDS]…`

> - The search is case insensitive. e.g `payment` will match `Payment`
> - The order of the keywords does not matter. e.g. `Payment Received` will match `Received Payment`
> - Only the name is searched
> - Only full words will be matched e.g. `Pay` will not match `Payment`
> - Items matching at least one keyword will be returned (i.e. `OR` search). e.g. `Received Payment` will return `Received Payment`, `Added Payment` and `Received Income`

Examples:

- `find-finance Payment`
  Returns `payment` and `Received Payment`
- `find-finance Course Student Teacher`
  Returns any finance having names `Course`, `Student`, or `Teacher`

5.11.5. List all finances: `list-finance`

Shows a list of all finances Format: `list-finance`

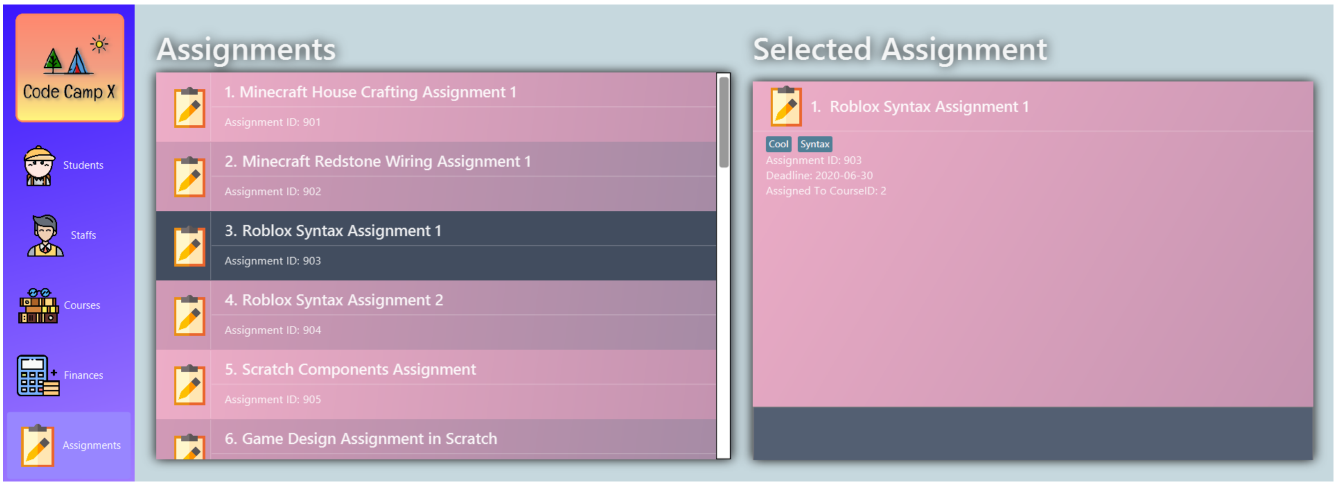
5.11.6. Get total earnings: `earnings`

Shows all the earnings (finances with positive amount) as a number.
Format: `earnings`

5.11.7. Get total expenses: `expenses`

Shows all the expenses (finances with positive amount) as a number.
Format: `expenses`

5.12. Assignment



Assignments

1. Minecraft House Crafting Assignment 1
Assignment ID: 901

2. Minecraft Redstone Wiring Assignment 1
Assignment ID: 902

3. Roblox Syntax Assignment 1
Assignment ID: 903

4. Roblox Syntax Assignment 2
Assignment ID: 904

5. Scratch Components Assignment
Assignment ID: 905

6. Game Design Assignment in Scratch

Selected Assignment

1. Roblox Syntax Assignment 1
Cool Syntax
Assignment ID: 903
Deadline: 2020-06-30
Assigned To CourseID: 2

Figure 10. List of Assignments. Shows details of selected Assignment.

5.12.1. Add an assignment: `add-assignment`

Add a new assignment to the list of all assignments
Format: `n/NAME dl/DEADLINE [t/TAG]…`
Example: * `add-assignment n/Adversarial Search Assignment 2 dl/2020-12-30 t/AI t/Difficult`

NOTE `DEADLINE` must be in `YYYY - MM - DD` or `YYY - MM - DD` format else an error will be thrown.

5.12.2. Delete an assignment: `delete-assignment`

Format: `delete-assignment ID`
Example:

- `delete-assignment 16100`

> - Deletes the item at the specified `ID`. The ID refers to the ID number shown in the displayed item list
> - The ID **must be a positive integer** 16100, 25200, 52500, …

NOTE Be reminded that when you delete an assignment, the assignment will be deleted from every course as well.
Furthermore, the progress of the assignment will also be removed from every course of any student
and every student of any course.
For example, a student that was only assigned to a course that contains only this assignment will have no assignment left after this assignment is deleted.

5.12.3. Edit an assignment: `edit-assignment`

Format: `edit-assignment ID [n/NAME] dl/DEADLINE t/TAGS]`
Example:

- `edit-assignment 16100 n/Edit Python Assignment 1`

- Edits the assignment at the specified `ID`. The ID refers to the ID number shown in the displayed course list panel
- The ID **must be a positive integer** 16100, 2520, 52500, …
- At least one of the optional fields must be provided
- Existing values will be updated to the input values
- When editing tags, the existing tags of the item will be removed i.e adding of tags is not cumulative
- You can remove all the assignment's tags by typing `t/` without specifying any tags after it

5.12.4. Locating items by name: `find-assignment`

Finds assignments whose names contain any of the given keywords.
Format: `find-assignment KEYWORD [MORE_KEYWORDS]…`

- The search is case insensitive. e.g `java` will match `Java`
- The order of the keywords does not matter. e.g. `Java Assignment` will match `Assignment Java`
- Only the name is searched
- Only full words will be matched e.g. `Java` will not match `Javascript`
- Items matching at least one keyword will be returned (i.e. `OR` search). e.g. `Java Assignment` will return `Java Course`, `Java Code` and `Python Assignment`

Examples:

- `find-assignment java`
  Returns `java` and `Java Assignment`

5.13. Upcoming Features

5.13.1. Add registration `[coming in v2.0]`

Add a `register` command by including role (i.e. privilege system).

Format: `register id/USERNAME pw/PASSWORD n/FULL_NAME r/ROLE_ID`

Examples:

- `register id/azhikai pw/1122qq n/Ang Zhi Kai r/999`

5.13.2. Creating user role `[coming in v2.0]`

Creates a user role. User assigned with higher ranking role can execute more commands.

Format: `create-role r/RANK n/ROLE_NAME`

Examples:

- `create-role r/999 n/Owner`
- `create-role r/2 n/Supervisor`
- `create-role r/1 n/Employee`

5.13.3. Editing user role `[coming in v2.0]`

Edits an existing user role.
Format: `edit-role r/RANK [nr/NEW_RANK] [n/ROLE_NAME]`

Examples:

- `edit-role r/999 n/Administrator`
- `edit-role r/2 nr/3 n/Manager`

5.13.4. Deleting user role `[coming in v2.0]`

Deletes an existing user role.
Format: `delete-role r/RANK`

Examples:

- `delete-role r/999`

6. Frequently Asked Questions (FAQ)

Q: How do I transfer my data to another Computer?
A: Install the application in the other computer and overwrite the empty data file it creates with the file that contains the data of your previous **CodeCampX** folder.

Q: Where do I find the latest release of the application?

A: You can find all releases here.

7. Command Summary

Table 1. Summary of command formats

Section	Command	Format	Example
Help	**Help**	`help`	`help`
Undo/Redo	**Undo an undoable command**	`undo`	`undo`
	Redo a redoable command	`redo`	`redo`
Select item	**Select a Student**	`select sid/STUDENTID`	`select sid/1`
	Select a Staff	`select tid/STAFFID`	`select tid/31`
	Select a Course	`select cid/COURSEID`	`select cid/1`
	Select Assignment belonging to the Course of a Student:	`select sid/STUDENTID cid/COURSEID`	`select sid/11 cid/1`
	Select Assignment belonging to the Student of a Course:	`select cid/COURSEID sid/STUDENTID`	`select cid/1 sid/11`
Switch tab	**Switch to Summary Tab**	`m` or `summary`	`m` or `summary`
	Switch to Student Tab	`s` or `student`	`s` or `student`
	Switch to Staff Tab	`t` or `staff`	`t` or `staff`
	Switch to Course Tab	`c` or `course`	`c` or `course`
	Switch to Finance Tab	`f` or `finance`	`f` or `finance`
	Switch to Assignment Tab	`a` or `assignment`	`a` or `assignment`

Section	Command	Format	Example
Assigning to Course	**Assign Student to a Course**	`assign cid/COURSEID sid/STUDENTID`	`assign cid/829 sid/33`
	Assign Teacher to a Course	`assign cid/COURSEID tid/TEACHERID`	`assign cid/829 tid/21`
	Assign Assignment to a Course	`assign cid/COURSEID aid/ASSIGNMENTID`	`assign cid/829 aid/21`
	Unassign Student from a Course	`unassign cid/COURSEID sid/STUDENTID`	`unassign cid/829 sid/33`
	Unassign Teacher from a Course	`unassign cid/COURSEID tid/TEACHERID`	`unassign cid/829 tid/21`
	Unassign Assignment from a Course	`unassign cid/COURSEID aid/ASSIGNMENTID`	`unassign cid/829 aid/21`
Done/Undone Commands	**Mark an Assignment as Done**	`done sid/STUDENTID aid/ASSIGNMENTID`	`done sid/33 aid/21`
	Mark an Assignment as Undone	`undone sid/STUDENTID aid/ASSIGNMENTID`	`undone sid/33 aid/21`
Student Commands	**Add Student**	`add-student n/NAME [t/TAG]…`	`add-student n/Jon Snow t/Hardworking`
	Delete Student	`delete-student ID`	`delete-student 16100`
	Find Student	`find-student KEYWORD [MORE_KEYWORDS]…`	`find-student Lannister Targaryen Stark`
	List Student	`list-student`	`list-student`
	Edit Student	`edit-student ID [n/NAME] [t/TAG]…`	`edit-student 16100 n/Aegon Targaryen t/knownothing`
Staff Commands	**Add Staff**	`add-staff n/NAME [t/TAG]…`	`add-staff n/Jon Snow t/Hardworking`
	Delete Staff	`delete-staff ID`	`delete-staff 16100`
	Find Staff	`find-staff KEYWORD [MORE_KEYWORDS]…`	`find-staff Lannister Targaryen Stark`
	List Staff	`list-staff`	`list-staff`
	Edit Staff	`edit-staff ID [n/NAME] [t/TAG]…`	`edit-staff 16100 n/Aegon Targaryen t/Son of Lyanna Stark and Rhaegar Targaryen`
Course Commands	**Add Course**	`add-course n/NAME a/AMOUNT [t/TAG]…`	`add-course n/Cozmo Programming a/2000 t/Fun t/Robot`
	Delete Course	`delete-course ID`	`delete-course 16100`
	Find Course	`find-course KEYWORD [MORE_KEYWORDS]…`	`find-course Java Python C`
	List Course	`list-course`	`list-course`
	Edit Course	`edit-course ID [n/NAME] [a/AMOUNT] [t/TAG]…`	`edit-course 16100 n/Java Programming a/2000`

Section	Command	Format	Example
Finance Commands	**Add Finance Type 1 (ft/m)**	`add-finance ft/FINANCETYPE d/DATE n/NAME a/AMOUNT [t/TAG]…`	`add-finance ft/m d/2020-12-09 n/Paid NTU a/1200 t/Partnership t/Monthly`
	Add Finance Type 2 (ft/cs)	`add-finance ft/FINANCETYPE d/DATE cid/COURSEID sid/STUDENTID [t/TAG]…`	`add-finance ft/cs d/2020-12-09 cid/829 sid/33 t/Late`
	Add Finance Type 3 (ft/ct)	`add-finance ft/FINANCETYPE d/DATE cid/COURSEID tid/TEACHERID [t/TAG]…`	`add-finance ft/ct d/2020-12-09 cid/829 tid/21 t/Early`
	Delete Finance	`delete-finance ID`	`delete-finance 16100`
	Find Finance	`find-finance KEYWORD [MORE_KEYWORDS]…`	`find-finance Lannister Targaryen Stark`
	List Finance	`list-finance`	`list-finance`
	Calculate Earnings	`earnings`	`earnings`
	Calculate Expenses	`expenses`	`expenses`