
UoL Grades Calculator

Release 0.5.2

Sebastien Lavoie

May 16, 2021

CONTENTS

1	Available commands	1
1.1	check	1
1.2	check score-accuracy	1
1.3	generate-sample	2
1.4	summarize	2
2	For developers	5
2.1	Getting a copy of the source code	5
2.2	Running the test suite	5
2.3	Developing locally as a package	5
2.4	Documentation	5
3	ugc package	7
3.1	Subpackages	7
3.2	Submodules	8
3.3	ugc.__main__ module	8
3.4	ugc.cli module	8
3.5	ugc.commands module	8
3.6	ugc.config module	8
3.7	ugc.errors module	9
3.8	ugc.grades module	9
4	Grades Calculator	11
5	Requirements	13
6	Install and uninstall	15
7	To run the utility	17
8	Generate a sample config file to get started	19
8.1	Specifying a different path for the config file	19
8.2	How to fill the config file (<i>.grades.yml</i> by default)	19
9	How to use this tool	21
10	Demo	23
	Python Module Index	25
	Index	27

AVAILABLE COMMANDS

1.1 check

```
$ ugc check --help
```

```
Usage: ugc check [OPTIONS] COMMAND [ARGS]...
```

```
Perform sanity checks against the results generated.
```

```
Options:
```

```
--help Show this message and exit.
```

```
Commands:
```

```
score-accuracy Check for rounding errors when averaging module score.
```

1.2 check score-accuracy

```
$ ugc check score-accuracy --help
```

```
Usage: ugc check score-accuracy [OPTIONS]
```

```
Check for rounding errors when averaging module score.
```

```
Options:
```

```
--help Show this message and exit.
```

Example output:

```
$ ugc check score-accuracy
```

```
Algorithms and Data Structures I: 78% actual [expected 79.0%]
```

```
Discrete Mathematics: 79.5% actual [expected 80.0%]
```

```
Fundamentals of Computer Science: 60% actual [expected 58.0%]
```

1.3 generate-sample

```
$ ugc generate-sample --help

Usage: ugc generate-sample [OPTIONS]

Generate a sample grades YAML config file.

Options:
-f, --force-overwrite  Overwrite the existing config file, if any.
--help                Show this message and exit.
```

Example output:

```
$ ugc generate-sample

Configuration file not found: /home/sglavoie/.grades.yml
→ Configuration file generated.
```

1.4 summarize

```
$ ugc summarize --help

Usage: ugc summarize [OPTIONS] COMMAND [ARGS]...

    Print a summary of the progress made so far.

Options:
  --help  Show this message and exit.

Commands:
  all      Output includes modules done as well as those in progress.
  done     Output includes only modules that are done and dusted.
  progress Output includes only modules that are in progress.
```

Example output:

```
$ ugc summarize done

Modules taken:
[ { 'Algorithms and Data Structures I': { 'completion_date': '2020-03',
                                          'final_score': 92,
                                          'final_weight': 50,
                                          'level': 4,
                                          'midterm_score': 98,
                                          'midterm_weight': 50,
                                          'module_score': 95}},
  { 'Discrete Mathematics': { 'completion_date': '2020-03',
                              'final_score': 100,
                              'final_weight': 50,
```

(continues on next page)

(continued from previous page)

```

        'level': 4,
        'midterm_score': 99,
        'midterm_weight': 50,
        'module_score': 100}},
{ 'Fundamentals of Computer Science': { 'completion_date': '2020-09',
        'final_score': 98,
        'final_weight': 50,
        'level': 4,
        'midterm_score': 98,
        'midterm_weight': 50,
        'module_score': 98}},
{ 'How Computers Work': { 'completion_date': '2018-12',
        'level': 4,
        'module_score': -1}},
{ 'Introduction to Programming I': { 'completion_date': '2019-09',
        'final_score': 100,
        'final_weight': 50,
        'level': 4,
        'midterm_score': 100,
        'midterm_weight': 50,
        'module_score': 100}},
{ 'Numerical Mathematics': { 'completion_date': '2019-09',
        'final_score': 61,
        'final_weight': 50,
        'level': 4,
        'midterm_score': 99,
        'midterm_weight': 50,
        'module_score': 80}},
{ 'Introduction to Programming II': { 'completion_date': '2020-03',
        'final_score': 98,
        'final_weight': 70,
        'level': 4,
        'midterm_score': 100,
        'midterm_weight': 30,
        'module_score': 99}},
{ 'Web Development': { 'completion_date': '2019-09',
        'final_score': 87,
        'final_weight': 70,
        'level': 4,
        'midterm_score': 86,
        'midterm_weight': 30,
        'module_score': 87}},
{ 'Algorithms and Data Structures II': { 'completion_date': '2020-09',
        'final_score': 92,
        'final_weight': 50,
        'level': 5,
        'midterm_score': 92,
        'midterm_weight': 50,
        'module_score': 92}},
{ 'Object Oriented Programming': { 'completion_date': '2020-09',
        'final_score': 96,
        'final_weight': 50,

```

(continues on next page)

(continued from previous page)

```
'level': 5,
'midterm_score': 96,
'midterm_weight': 50,
'module_score': 96}}]

Number of modules done: 10
Scores so far: [95, 100, 98, 100, 80, 99, 87, 92, 96]

Weighted average: 94.08
ECTS: A
US: A

Unweighted average: 94.11
ECTS: A
US: A

Classification (weighted): First Class Honours

ECTS grade equivalence:
{ 'Algorithms and Data Structures I': 'A',
  'Algorithms and Data Structures II': 'A',
  'Discrete Mathematics': 'A',
  'Fundamentals of Computer Science': 'A',
  'How Computers Work': 'N/A',
  'Introduction to Programming I': 'A',
  'Introduction to Programming II': 'A',
  'Numerical Mathematics': 'A',
  'Object Oriented Programming': 'A',
  'Web Development': 'A'}

US grade equivalence:
{ 'Algorithms and Data Structures I': 'A',
  'Algorithms and Data Structures II': 'A-',
  'Discrete Mathematics': 'A',
  'Fundamentals of Computer Science': 'A',
  'How Computers Work': 'N/A',
  'Introduction to Programming I': 'A',
  'Introduction to Programming II': 'A',
  'Numerical Mathematics': 'B-',
  'Object Oriented Programming': 'A',
  'Web Development': 'B+'}

GPA (weighted): 4 US - 4 UK
Total credits done: 150 / 360 (41.67%)
```


FOR DEVELOPERS

2.1 Getting a copy of the source code

Clone this repository.

2.2 Running the test suite

```
$ pip install -r requirements-dev.txt
$ pytest
```

2.3 Developing locally as a package

Installing the necessary requirements:

```
$ pip install -r requirements.txt
```

Building the application once (no need to rebuild to test changes on the source code):

```
$ python setup.py develop
```

Then the command `ugc` (short for `uol grades calculator`) becomes available on the command-line. Type `ugc --help` for more information.

The tool can then be uninstalled using the following command:

```
$ python setup.py develop --uninstall
```

2.4 Documentation

2.4.1 Generating modules documentation

```
$ cd docs/
$ make docs
```

Table 1: Current options passed to build the docs

Flag	Description
-f	overwrite existing files
-M	put module documentation before submodule
-P	include “_private” modules
-o	output directory (docs/source/)
-d	maximum depth of submodules to show in the TOC (set to 1)
-T	do not add a TOC for the modules

2.4.2 Rebuilding documentation

```
$ cd docs/  
$ make html
```

If something is not rendered even after a force-refresh, try running `make clean html` instead: there can be instances where changes are not applied due to the local cache.

UGC PACKAGE

3.1 Subpackages

3.1.1 ugc.utils package

Submodules

ugc.utils.commands_helpers module

`ugc.utils.commands_helpers.get_module_score(module)` → float

`ugc.utils.commands_helpers.get_module_score_rounded_up(module)` → float

ugc.utils.grades_helpers module

`ugc.utils.grades_helpers.get_classification(average)` → str

Return a string containing the classification of the student according to the Programme Specification.

`ugc.utils.grades_helpers.get_ects_equivalent_score(score: int)` → str

Return the grade in the ECTS equivalent form. Range from A to E/F.

`ugc.utils.grades_helpers.get_score_of_module_in_progress(module: dict)` → float

`ugc.utils.grades_helpers.get_total_score_modules_finished(modules: list)` → float

`ugc.utils.grades_helpers.get_total_score_modules_in_progress(modules: list)` → float

`ugc.utils.grades_helpers.get_total_weight_modules_finished(modules: list)` → float

`ugc.utils.grades_helpers.get_total_weight_modules_in_progress(modules: list)` → float

`ugc.utils.grades_helpers.get_uk_gpa(average)` → float

Return the GPA as calculated in the UK.

`ugc.utils.grades_helpers.get_us_gpa(average)` → float

Return the GPA as calculated in the US.

`ugc.utils.grades_helpers.get_us_letter_equivalent_score(score: float)` → str

Get the letter equivalent in the US grading system for a given score.

`ugc.utils.grades_helpers.get_weight_of(level: int)` → int

Return the weight of a given *level*. The ratio is 1:3:5 for modules of L4:L5:L6 respectively.

`ugc.utils.grades_helpers.score_is_valid(module_score: float)` → bool

Check whether a given score is a valid numeric value. Return a Boolean value.

ugc.utils.mathtools module

Math helper functions.

`ugc.utils.mathtools.round_half_up(num: float, decimals=0)`
Round a float up and return it.

3.2 Submodules

3.3 ugc.__main__ module

Allows calling `python -m ugc` from the root of the project.

3.4 ugc.cli module

Describes the commands available from the terminal when running this tool.

3.5 ugc.commands module

List the commands available from the CLI: one per function.

`ugc.commands.check_score_accuracy(grades) → dict`

`ugc.commands.generate_sample(config, force_overwrite=False) → bool`
Generate a sample grades YAML config file.

`ugc.commands.summarize_all(grades: object, symbol: str = '=', repeat: int = 60) → None`
Print a summary of modules done and in progress.

`ugc.commands.summarize_done(grades)`
Print a summary of the progress made so far for modules that are done and dusted.

`ugc.commands.summarize_progress(grades)`
Print a summary of only the modules that are currently in progress.

3.6 ugc.config module

Manage the configuration file.

`class ugc.config.Config(config_path=None)`
Bases: object

Loads the configuration file where grades are stored.

`static check_total_weight_sums_up_100_for_module(module, module_name) → bool`

`check_total_weight_sums_up_100_in_all_modules() → bool`

`load() → None`
Load grades from a YAML file.

verify() → bool

Check that the config file contains valid data. Return True when it's valid, False otherwise.

3.7 ugc.errors module

exception `ugc.errors.ConfigValidationError(custom_msg)`

Bases: `Exception`

Raised when there is an error in the config file.

3.8 ugc.grades module

Command-line application to get information about progress made in a BSc Computer Science at the University of London (calculations are specific to this particular degree).

class `ugc.grades.Grades(config_path=None)`

Bases: `object`

calculate_unweighted_average() → float

Return the unweighted average across all completed modules.

calculate_unweighted_average_in_progress() → float

Return the unweighted average across all completed modules and those in progress.

calculate_weighted_average() → float

calculate_weighted_average_in_progress() → float

get_list_of_finished_modules() → list

Return a list of dicts containing information about all the modules that have a valid score (either -1 or 0 ≤ x ≤ 100).

get_list_of_modules_in_progress() → list

Return a list of dict containing all the non-empty values of the modules in progress.

get_module_scores_of_finished_modules() → list

Return a list of floats with the score obtained in each module.

get_module_scores_of_finished_modules_for_system(*system: str = 'US'*) → dict

Return a dictionary containing the converted ECTS score for each module.

get_num_of_finished_modules() → int

Return the number of modules completed with a score greater than or equal to zero as an integer.

get_percentage_degree_done() → float

From the total number of credits, return the percentage done out of 360 credits.

get_scores_of_modules_in_progress() → list

Return a list of floats with the score obtained in each module in progress.

get_scores_of_modules_in_progress_for_system(*system: str = 'US'*) → dict

Return a dictionary containing the converted ECTS score for each module in progress.

get_total_credits() → int

Get the total number of credits gotten so far as an integer.

load() → None

Perform basic calculations required for most commands.

GRADES CALCULATOR

This tool is all about getting information and generating insights from the progress made in a [BSc Computer Science at the University of London](#) (calculations are specific to this particular degree).

REQUIREMENTS

Python 3.6 and above. This is it!

INSTALL AND UNINSTALL

The most straightforward way to use this tool would be to install it from [PyPI](#) by typing the following in a terminal (use of [virtual environment](#) recommended!):

```
$ pip install uol-grades-calculator
```

Reversing the process is a matter of typing this:

```
$ pip uninstall uol-grades-calculator
```


TO RUN THE UTILITY

```
$ ugc
```

By passing no arguments, this will print the default help message.

GENERATE A SAMPLE CONFIG FILE TO GET STARTED

To generate a sample configuration file, run the following command:

```
$ ugc generate-sample
```

The configuration file will be created in your home directory as a hidden file (i.e. `~/.grades.yml`).

8.1 Specifying a different path for the config file

If you want to create it somewhere else:

```
$ ugc --config /path/to/config/file.yml generate-sample
```

Note that you will have to indicate where the config is each time you use this tool in this case (you can always create an alias to avoid the trouble of typing it every time). For example:

```
$ ugc --config /path/to/config/file.yml summarize
```

8.2 How to fill the config file (`.grades.yml` by default)

Each module described in the config file should contain information adhering to the following indications:

Table 1: Configuration options

YAML node	Value	Example(s)	Optional*
<code>completion_date</code>	Date as a string : <i>YYYY-MM</i>	2020-01	Yes
<code>final_score</code>	Float : <i>range</i> 0.00–100.00	50, 50.5, 90.56	Yes
<code>final_weight</code>	Integer expressing a percentage: <i>range</i> 0–100	0, 40, 80, 100	Yes
<code>midterm_score</code>	Float : <i>range</i> 0.00–100.00	50, 50.5, 90.56	Yes
<code>midterm_weight</code>	Integer expressing a percentage: <i>range</i> 0–100	0, 40, 80, 100	Yes
<code>module_score</code>	Float : <i>range</i> 0.00–100.00	50, 50.5, 90.56	No
<code>level</code>	Integer : choose <i>strictly</i> from 4, 5 or 6	4, 5, 6	No

* If a node value is left empty (or the node is absent in a given module), this will affect how the module is taken into account (average across all modules, summary of modules taken, etc.).

Here is a complete example for one module:

```
Algorithms and Data Structures I:  
  completion_date: 2020-03  
  final_score: 92  
  final_weight: 50  
  midterm_score: 98  
  midterm_weight: 50  
  module_score: 95  
  level: 4
```

8.2.1 Module taken

This means we define a module score between *0* and *100*, both being inclusive values.

```
Algorithms and Data Structures I:  
  module_score: 80.5
```

8.2.2 Module recognized (RPL)

In this case, we define a score of *-1* to indicate that this module is done but we didn't get a score for it.

```
Algorithms and Data Structures I:  
  module_score: -1
```


HOW TO USE THIS TOOL

Please refer to the page [Available commands](#) to see what ugc can do for you.

CHAPTER
TEN

DEMO

PYTHON MODULE INDEX

U

- [ugc](#), 7
- [ugc.__main__](#), 8
- [ugc.cli](#), 8
- [ugc.commands](#), 8
- [ugc.config](#), 8
- [ugc.errors](#), 9
- [ugc.grades](#), 9
- [ugc.utils](#), 7
- [ugc.utils.commands_helpers](#), 7
- [ugc.utils.grades_helpers](#), 7
- [ugc.utils.mathtools](#), 8

C

calculate_unweighted_average() (ugc.grades.Grades method), 9
 calculate_unweighted_average_in_progress() (ugc.grades.Grades method), 9
 calculate_weighted_average() (ugc.grades.Grades method), 9
 calculate_weighted_average_in_progress() (ugc.grades.Grades method), 9
 check_score_accuracy() (in module ugc.commands), 8
 check_total_weight_sums_up_100_for_module() (ugc.config.Config static method), 8
 check_total_weight_sums_up_100_in_all_modules() (ugc.config.Config method), 8
 Config (class in ugc.config), 8
 ConfigValidationError, 9

G

generate_sample() (in module ugc.commands), 8
 get_classification() (in module ugc.utils.grades_helpers), 7
 get_ects_equivalent_score() (in module ugc.utils.grades_helpers), 7
 get_list_of_finished_modules() (ugc.grades.Grades method), 9
 get_list_of_modules_in_progress() (ugc.grades.Grades method), 9
 get_module_score() (in module ugc.utils.commands_helpers), 7
 get_module_score_rounded_up() (in module ugc.utils.commands_helpers), 7
 get_module_scores_of_finished_modules() (ugc.grades.Grades method), 9
 get_module_scores_of_finished_modules_for_system() (ugc.grades.Grades method), 9
 get_num_of_finished_modules() (ugc.grades.Grades method), 9
 get_percentage_degree_done() (ugc.grades.Grades method), 9
 get_score_of_module_in_progress() (in module ugc.utils.grades_helpers), 7

get_scores_of_modules_in_progress() (ugc.grades.Grades method), 9
 get_scores_of_modules_in_progress_for_system() (ugc.grades.Grades method), 9
 get_total_credits() (ugc.grades.Grades method), 9
 get_total_score_modules_finished() (in module ugc.utils.grades_helpers), 7
 get_total_score_modules_in_progress() (in module ugc.utils.grades_helpers), 7
 get_total_weight_modules_finished() (in module ugc.utils.grades_helpers), 7
 get_total_weight_modules_in_progress() (in module ugc.utils.grades_helpers), 7
 get_uk_gpa() (in module ugc.utils.grades_helpers), 7
 get_us_gpa() (in module ugc.utils.grades_helpers), 7
 get_us_letter_equivalent_score() (in module ugc.utils.grades_helpers), 7
 get_weight_of() (in module ugc.utils.grades_helpers), 7
 Grades (class in ugc.grades), 9

L

load() (ugc.config.Config method), 8
 load() (ugc.grades.Grades method), 9

M

module
 ugc, 7
 ugc.__main__, 8
 ugc.cli, 8
 ugc.commands, 8
 ugc.config, 8
 ugc.errors, 9
 ugc.grades, 9
 ugc.utils, 7
 ugc.utils.commands_helpers, 7
 ugc.utils.grades_helpers, 7
 ugc.utils.mathtools, 8

R

round_half_up() (in module ugc.utils.mathtools), 8

S

`score_is_valid()` (in *module*
ugc.utils.grades_helpers), 7
`summarize_all()` (in *module* *ugc.commands*), 8
`summarize_done()` (in *module* *ugc.commands*), 8
`summarize_progress()` (in *module* *ugc.commands*), 8

U

`ugc`
 module, 7
`ugc.__main__`
 module, 8
`ugc.cli`
 module, 8
`ugc.commands`
 module, 8
`ugc.config`
 module, 8
`ugc.errors`
 module, 9
`ugc.grades`
 module, 9
`ugc.utils`
 module, 7
`ugc.utils.commands_helpers`
 module, 7
`ugc.utils.grades_helpers`
 module, 7
`ugc.utils.mathtools`
 module, 8

V

`verify()` (*ugc.config.Config* *method*), 8