# Level 2 Allocation (Assignment 1)

Generated by Doxygen 1.8.13

# Contents

# Chapter 1

# File Index

## 1.1 File List

Here is a list of all documented files with brief descriptions:

# Chapter 2

# File Documentation

## 2.1 src/CalcModule.py File Reference

This file contains functions that manipulates data provided by the ReadAllocationData file.

### Functions

- def CalcModule.sort (S)

  *Sorts a list of dictionaries by their 'gpa' in descending order.*
- def CalcModule.average (L, g)

  *Calculates the average GPA of a specified gender.*
- def CalcModule.allocate (S, F, C)

  *Allocates students into an engineering stream.*

### 2.1.1 Detailed Description

This file contains functions that manipulates data provided by the ReadAllocationData file.

**Author**

Zayed Sheet

**Date**

2019-01-18

### 2.1.2 Function Documentation

#### 2.1.2.1 allocate()

```
def CalcModule.allocate (
              S,
              F,
              C )
```

Allocates students into an engineering stream.

The function sorts a list of students by their GPA in descending order. It then adds everyone with a GPA below 4.0 into a text file and allocates everyone with free choice into their first choice. The program then allocates everyone else from highest to lowest gpa into their first, second or third choice in that order depending on which one has available capacity first. Anyone that isn't allocated into a program (because all three capacities are full) gets placed into a text file

**Parameters**

| | |
|---|---|
| *S* | This parameter is a list of dictionaries where each dictionary represents a student with several attributes (such as) name, macid, program choices, etc.) |
| *F* | This parameter is a list of students that have free choice. Each student is represented by their macid. |
| *C* | This parameter is a dictionary with engineering departments as keys and their respective capacity as the key's value. |

**Returns**

This function returns a dictionary that contains the engineering streams as keys and a list of dictionaries as values. Each dictionary represents a student that is allocated into that engineering stream.

#### 2.1.2.2 average()

```
def CalcModule.average (
              L,
              g )
```

Calculates the average GPA of a specified gender.

The function adds together the floating point values for all dictionaries that have the specified 'gender' value that was passed into the function. While doing this it also counts how many dictionaries from the list have that gender value. It then divides the total gpa by the counter to find the average.

**Parameters**

| | |
|---|---|
| *L* | This parameter is a list of dictionaries that have a 'gpa' and 'gender' key. |
| *g* | This parameter is a string for the gender you want to find the average for. |

**Returns**

The returned value is a value rounded to two decimal places for the average GPA of a specified gender.

#### 2.1.2.3 sort()

```
def CalcModule.sort (
              S )
```

Sorts a list of dictionaries by their 'gpa' in descending order.

This function sorts a list of dictionaries by the value of their 'gpa' key. of this code is referenced from 'https://stackoverflow.com/questions/72899/how-do-i-sort-a-list-of-dictionaries-by-a-valu

**Parameters**

| | |
|---|---|
| *S* | The single parameter this function takes is a list of dictionaries, each containing a 'gpa' key. |

**Returns**

The returned value is a new list of dictionaries, sorted in descending order by the value of thier 'gpa' keys.

## 2.2 src/ReadAllocationData.py File Reference

This file contains functions that extracts student/department data from text files.

### Functions

- def ReadAllocationData.readStdnts (s)
  *Uses the data from a text file to create a list of dictionaries.*
- def ReadAllocationData.readFreeChoice (s)
  *Obtains the macids of everyone that has free choice from a text file.*
- def ReadAllocationData.readDeptCapacity (s)
  *Obtains information on the engineering streams and their respective capacities.*

### 2.2.1 Detailed Description

This file contains functions that extracts student/department data from text files.

**Author**

Zayed Sheet

**Date**

2019-01-18

### 2.2.2 Function Documentation

#### 2.2.2.1 readDeptCapacity()

```
def ReadAllocationData.readDeptCapacity (
              s )
```

Obtains information on the engineering streams and their respective capacities.

**Parameters**

| | |
|---|---|
| *s* | This parameter is a text file that contains the engineering streams and their respective capacities. line in the text file should be formatted as such: 'program: capacity' |

**Returns**

Returns a dictionary where each key is an engineering stream and the value is its' respective capacity.

#### 2.2.2.2 readFreeChoice()

```
def ReadAllocationData.readFreeChoice (
              s )
```

Obtains the macids of everyone that has free choice from a text file.

**Parameters**

| | |
|---|---|
| *s* | This parameter is a text file that has the macids of everyone with free choice. Each line should be an individual macid in the text file. |

**Returns**

Returns a list of macids for students that have free choice.

#### 2.2.2.3 readStdnts()

```
def ReadAllocationData.readStdnts (
              s )
```

Uses the data from a text file to create a list of dictionaries.

Reads from a text file and uses the data to create a list of dictionaries where each dictionary represents a student. of the code for this function was referenced from:
https://stackoverflow.com/questions/4842057/easiest-way-to-ignore-blank-lines-when-readi

**Parameters**

| | |
|---|---|
| *s* | This parameter is a text file that contains a student on each line. Each line should contain the student's macid firstname, lastname, gender, gpa and top 3 engineering stream choices. line in the textfile should be formatted as such: 'macid, firstname, lastname, gender(male/female), gpa, firstchoice, secondchoice, thirdchoice' |

**Returns**

This function returns a list of dictionaries where each dictionary represents a student with their macid, fname, lname, gpa, choices as keys. The choices key contains a list with their top 3 engineering stream choices.

# Index