
CMPE 230 PROJECT DESCRIPTION

Ahmet Emir Kocaağa, Alperen Bağ
2017400276, 2018400279

Boun Schedule Scraper

Introduction

This program crawls Bogazici University's OBIKAS registration pages and extracts the necessary information for any semester between 1998 Fall and 2019 Spring. It reads the course offerings of each department for the given semesters and gives a csv table as a standart output. The table contains the Department Names, Course Codes, Course Names, which semesters that courses are offered and total number of graduate, under graduate courses and number of distinct instructors for each semesters and also for all of the given semesters totally.

How to Use?

1. Run the program with giving the desired range of semesters as an argument such as start[1] and end[2] semester.

For example: `>./bucourses.py 2018-Fall 2019-Spring`

Implementation Details

1- get_semester_column_name(semester)

This function gets a semester and converts it to a string(semester name) such as (2017, 2018, 1) to 2017-Fall for writing to heads of columns.

2- get_semester_name(str)

This function gets a semester name and converts it to a semester such as 2017-Fall to (2017, 2018, 1) for converting to given input to numbers.

3- get_course_name(str)

This function takes a course and department name as parameter, convert it to human readable form by getting rid of '+', '%26', '%3a', '%2c'.

4- course_statistics(dept, semester)

This function calculates the number of graduate courses, undergraduate courses and distinct instructors of a department in a semester.

5- course_statistics_total(dept)

This function takes all_courses dictionary and calculates number of graduate and undergraduate courses that is offered in all semesters for given department.

6- total_offerings_func(dept)

This function takes all_courses dictionary and calculates number of graduate, undergraduate courses and distinct instructors for a department for total offerings part.

7- Main Part of The Program

In this part, firstly we added the list of department codes and department names. Then, a list of all possible semesters are created. After that, inputs are taken and those inputs are converted to desired form for our program by get_semester_name function. By these inputs, we find the necessary part of the semesters list and also we create a list that contains names of these necessary semesters by get_semester_column_name function.

In short, our program iterates over the semesters that are in range which is specified by the terminal argument and it also iterates over the all departments for each semester. In one iteration, it fetches the html table as pandas.DataFrame from the url that is builded by using department and semester variables. After fetching, the dictionary named all_courses is updated according to the new information. The keys of all_courses are tuples that contain the short and long name of departments. The values of all_courses are also a dictionary whose keys are course codes. For example,

```
all_courses[("CMPE", "COMPUTER+ENGINEERING")] = {"CMPE150": a list, "CMPE160": a list}
```

The values indicated by course codes are lists which store the course information. The information is stored as following structure;

```
[department of the course, course name, instructor list by semester, total offering number]
```

Rest of the code is for building the dataframe that will be printed out in csv format. For each course in `all_courses`, a row is created and above mentioned functions are used to fill that rows with the information of the number of undergrad, grad and distinct instructors.

8- Printing Out the Data

We use the `StringIO` class located in `io` library in order to print the dataframe out in csv format on console. To convert the dataframe to a printable csv format, we use `pandas.to_csv()` function. As parameter, we give `StringIO` object to `to_csv()` function, then `StringIO` object is printed.