CMPE 230 – SYSTEMS PROGRAMMING
Project 2

Author: **Ali BATIR**
Student ID: **2015400261**

Author: **Kaan ŞENPARLAK**
Student ID: **2016401189**

Introduction

In this project, we developed a Python program called **bucourses.py** that will crawl Bogazici University's OBIKAS registration pages and extract course offering information.

By changing the semester and department fields, we access course offering information for different semesters and departments. This python program crawl course offerings of all department/ programs for all semesters given in a range and output the information given in the next page in CSV table format.

Example:

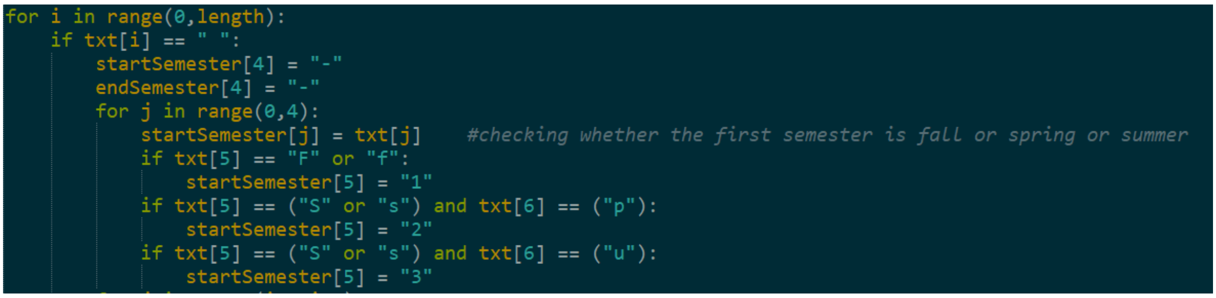
```
> ./bucourses.py 2018-Fall 2019-Spring
```

Dept./Prog. (name)	Course Code	Course Name	2018-Fall	2019-Spring	Total Offerings
...					
...					
ELKT (Electrical Eng.)	U4 G2		U3 G2 I4	U2 G2 I3	U5 G4 I5
	ELKT112	coursename		x	1 / 1
	ELKT213	coursename	x	x	2 / 1
	ELKT316	coursename	x		1 / 1
	ELKT451	coursename	x		1 / 1
	ELKT513	coursename	x	x	2 / 2
	ELKT618	coursename	x	x	2 / 1
....					
....					
KMPE (Computer Eng.)	U3 G2		U2 G2 I4	U2 G1 I3	U4 G3 I4
	KMPE150	coursename	x	x	2 / 2
	KMPE230	coursename		x	1 / 1
	KMPE423	coursename	x		1 / 1
	KMPE512	coursename	x	x	2 / 1
	KMPE530	coursename	x		1 / 1
...					
...					

Implementation

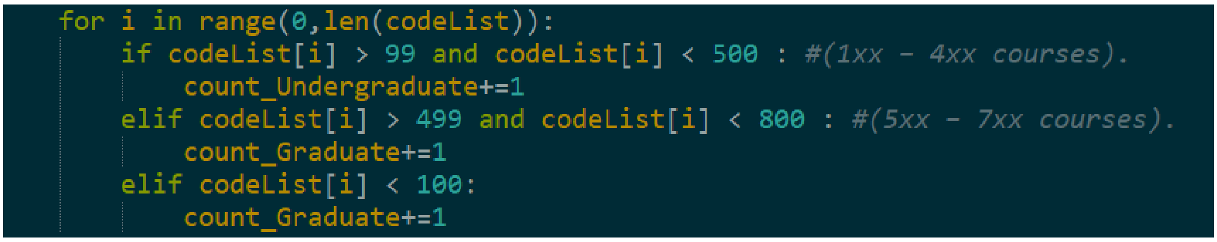
In this example, input corresponds to two semesters. (2018-Fall 2019-Spring).We get all terms between these semesters and get all the data of the departments. Firstly, we check whether the first semester is fall, spring or summer.

If term is Fall,then we understand that this term is “1”, if term is Spring,then we understand that this term is “2” and if term is Summer,then we understand that this term is “3”.



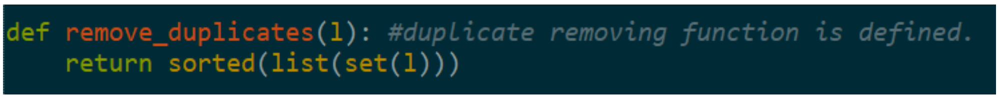
```
for i in range(0,length):
    if txt[i] == " ":
        startSemester[4] = "-"
        endSemester[4] = "-"
        for j in range(0,4):
            startSemester[j] = txt[j]    #checking whether the first semester is fall or spring or summer
            if txt[5] == "F" or "f":
                startSemester[5] = "1"
            if txt[5] == ("S" or "s") and txt[6] == ("p"):
                startSemester[5] = "2"
            if txt[5] == ("S" or "s") and txt[6] == ("u"):
                startSemester[5] = "3"
```

We get code sections and split them in order to take only numbers of the courses.If course numbers are bigger 99 and lower 500,then we increase the variable (count_Undergraduate) 1 but if courses codes are bigger 499 and lower 800 or lower 100,we increase the variable(count_Graduate) 1.



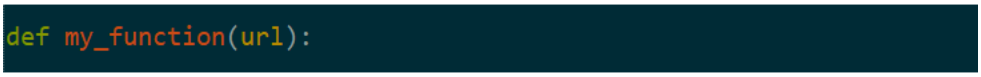
```
    for i in range(0,len(codeList)):
        if codeList[i] > 99 and codeList[i] < 500 : #(1xx - 4xx courses).
            count_Undergraduate+=1
        elif codeList[i] > 499 and codeList[i] < 800 : #(5xx - 7xx courses).
            count_Graduate+=1
        elif codeList[i] < 100:
            count_Graduate+=1
```

This function take a list and convert it to set,this provide remove same elements.Then,sorted this list and return.



```
def remove_duplicates(l): #duplicate removing function is defined.
    return sorted(list(set(l)))
```

This function get links,pulling data out of html and return required data.

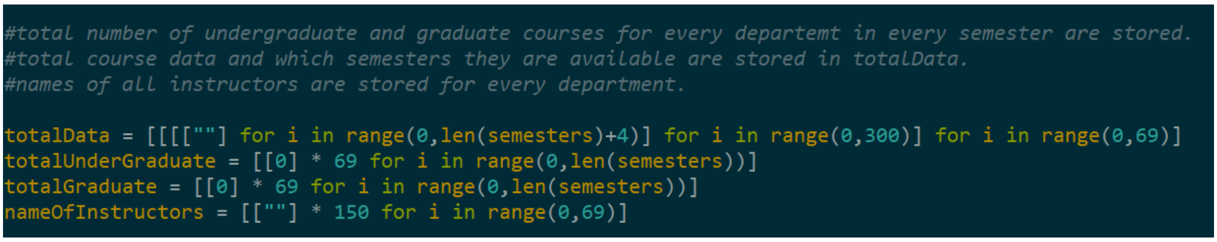


```
def my_function(url):
```

Count_Instructor is a list and keep the number of Instructors.Count_Graduate is a list and keep the number of graduate courses. Count_Undergraduate is a list and keep the number of undergraduate courses.Courses a multidimensional list keep data in this way;

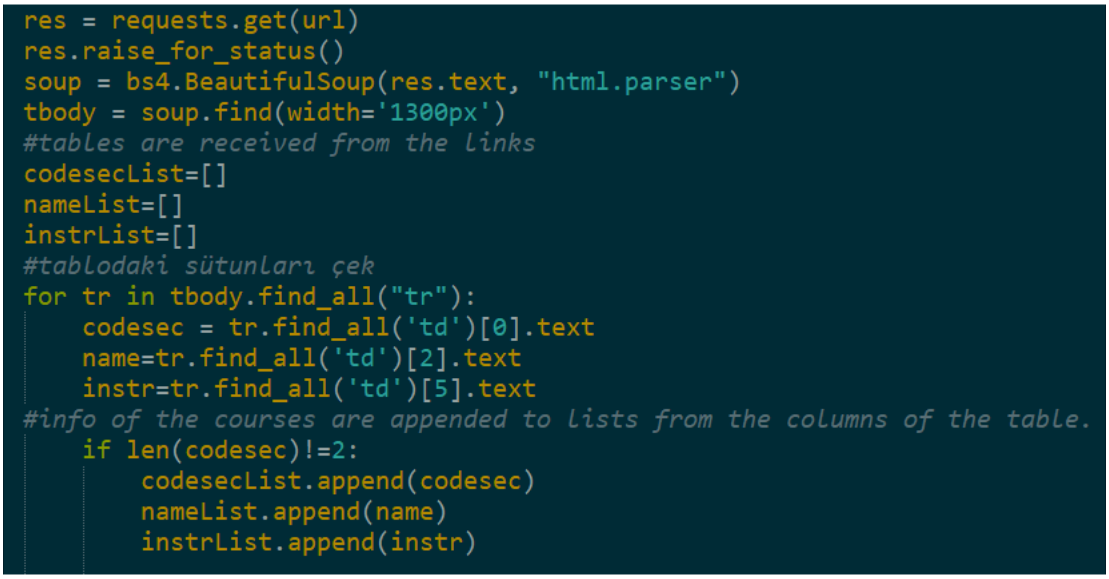
Courses[semester][department][course][other data]

other data; If it is [0], then we get course code.If it is [1],then we get course name. If it is [2], then we get instructor.Finally,if it is [3],then we get the course graduate or undergraduate.



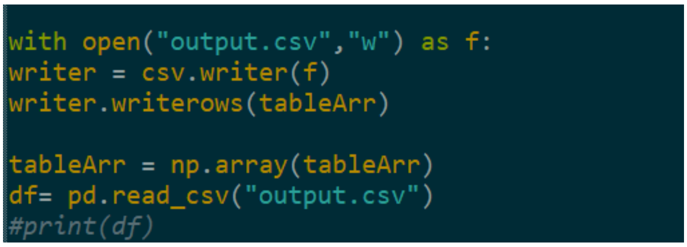
```
#total number of undergraduate and graduate courses for every departemt in every semester are stored.
#total course data and which semesters they are available are stored in totalData.
#names of all instructors are stored for every department.

totalData = [[[[""] for i in range(0,len(semesters)+4)] for i in range(0,300)] for i in range(0,69)]
totalUnderGraduate = [[0] * 69 for i in range(0,len(semesters))]
totalGraduate = [[0] * 69 for i in range(0,len(semesters))]
nameOfInstructors = [[""] * 150 for i in range(0,69)]
```

We use beautifisoup4 library. Beautiful Soup is a Python library for pulling data out of HTML and XML files.We get data in the table and create lists in order to keep data.We find all rows and columns and just take only necessary columns.



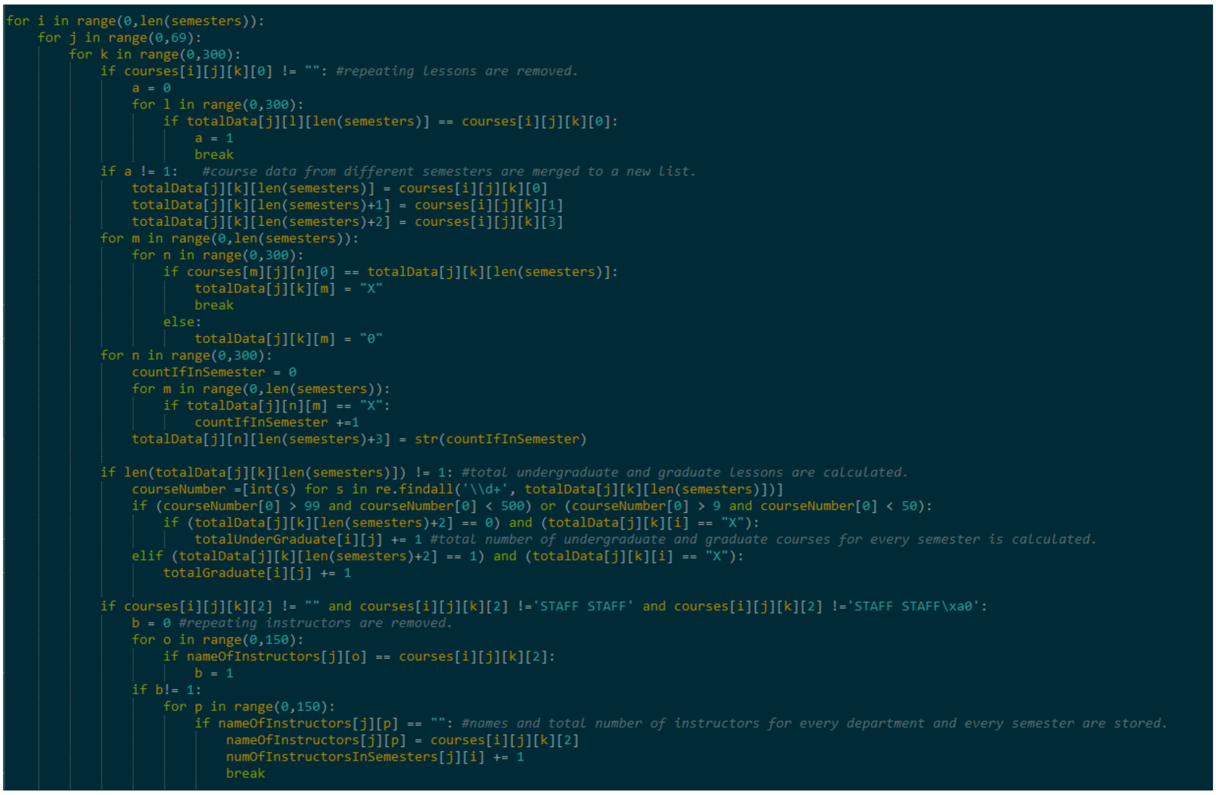
```
res = requests.get(url)
res.raise_for_status()
soup = bs4.BeautifulSoup(res.text, "html.parser")
tbody = soup.find(width='1300px')
#tables are received from the links
codesecList=[]
nameList=[]
instrList=[]
#tablodaki sütunları çek
for tr in tbody.find_all("tr"):
    codesec = tr.find_all('td')[0].text
    name=tr.find_all('td')[2].text
    instr=tr.find_all('td')[5].text
#info of the courses are appended to lists from the columns of the table.
    if len(codesec)!=2:
        codesecList.append(codesec)
        nameList.append(name)
        instrList.append(instr)
```

We use csv file in order to give output. A csv file is a type of plain text file that uses specific structuring to arrange tabular data. CSV files use a comma to separate each specific data value.We create a csv file called output.csv and we insert the data in the list(tableArr) to the file.



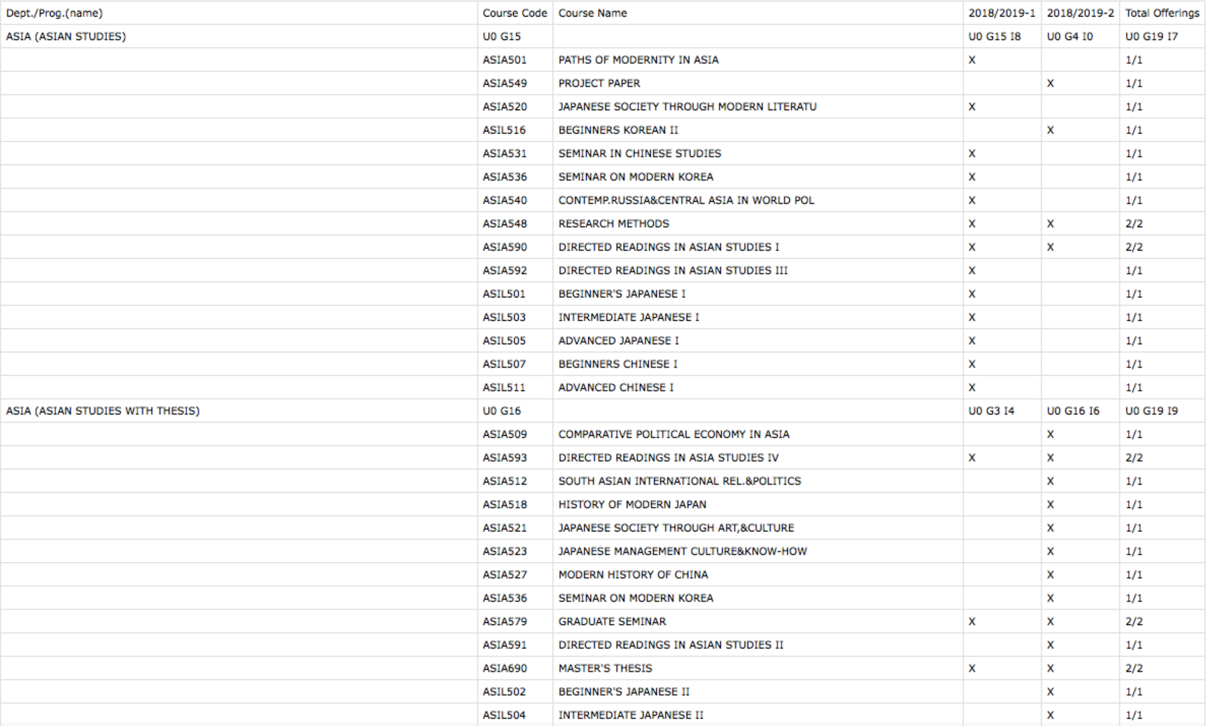
```
with open("output.csv","w") as f:
writer = csv.writer(f)
writer.writerows(tableArr)

tableArr = np.array(tableArr)
df= pd.read_csv("output.csv")
#print(df)
```

In this section of the code the data crawled from the links are manipulated in order to make it suitable for the csv format. totalData holds all the courses and whether they are offered in semesters and if undergrad or grad. Total number of graduate and undergraduate courses for every semester is stored in a list. Total number of instructors for every semester is also stored in another array.



```
for i in range(0,len(semesters)):
    for j in range(0,69):
        for k in range(0,300):
            if courses[i][j][k][0] != "": #repeating lessons are removed.
                a = 0
                for l in range(0,300):
                    if totalData[j][l][len(semesters)] == courses[i][j][k][0]:
                        a = 1
                        break
            if a != 1:   #course data from different semesters are merged to a new list.
                totalData[j][k][len(semesters)] = courses[i][j][k][0]
                totalData[j][k][len(semesters)+1] = courses[i][j][k][1]
                totalData[j][k][len(semesters)+2] = courses[i][j][k][3]
            for m in range(0,len(semesters)):
                for n in range(0,300):
                    if courses[m][j][n][0] == totalData[j][k][len(semesters)]:
                        totalData[j][k][m] = "X"
                        break
                    else:
                        totalData[j][k][m] = "0"
            for n in range(0,300):
                countIfInSemester = 0
                for m in range(0,len(semesters)):
                    if totalData[j][n][m] == "X":
                        countIfInSemester +=1
                totalData[j][n][len(semesters)+3] = str(countIfInSemester)

            if len(totalData[j][k][len(semesters)]) != 1: #total undergraduate and graduate lessons are calculated.
                courseNumber =[int(s) for s in re.findall('\\d+', totalData[j][k][len(semesters)])]
                if (courseNumber[0] > 99 and courseNumber[0] < 500) or (courseNumber[0] > 9 and courseNumber[0] < 50):
                    if (totalData[j][k][len(semesters)+2] == 0) and (totalData[j][k][i] == "X"):
                        totalUnderGraduate[i][j] += 1 #total number of undergraduate and graduate courses for every semester is calculated.
                elif (totalData[j][k][len(semesters)+2] == 1) and (totalData[j][k][i] == "X"):
                    totalGraduate[i][j] += 1

            if courses[i][j][k][2] != "" and courses[i][j][k][2] !='STAFF STAFF' and courses[i][j][k][2] !='STAFF STAFF\xa0':
                b = 0 #repeating instructors are removed.
                for o in range(0,150):
                    if nameOfInstructors[j][o] == courses[i][j][k][2]:
                        b = 1
                if b!= 1:
                    for p in range(0,150):
                        if nameOfInstructors[j][p] == "": #names and total number of instructors for every department and every semester are stored.
                            nameOfInstructors[j][p] = courses[i][j][k][2]
                            numOfInstructorsInSemesters[j][i] += 1
                            break
```

Output Example :



Dept./Prog.(name)	Course Code	Course Name	2018/2019-1	2018/2019-2	Total Offerings
ASIA (ASIAN STUDIES)	U0 G15		U0 G15 I8	U0 G4 I0	U0 G19 I7
	ASIA501	PATHS OF MODERNITY IN ASIA	X		1/1
	ASIA549	PROJECT PAPER		X	1/1
	ASIA520	JAPANESE SOCIETY THROUGH MODERN LITERATU	X		1/1
	ASIL516	BEGINNERS KOREAN II		X	1/1
	ASIA531	SEMINAR IN CHINESE STUDIES	X		1/1
	ASIA536	SEMINAR ON MODERN KOREA	X		1/1
	ASIA540	CONTEMP.RUSSIA&CENTRAL ASIA IN WORLD POL	X		1/1
	ASIA548	RESEARCH METHODS	X	X	2/2
	ASIA590	DIRECTED READINGS IN ASIAN STUDIES I	X	X	2/2
	ASIA592	DIRECTED READINGS IN ASIAN STUDIES III	X		1/1
	ASIL501	BEGINNER'S JAPANESE I	X		1/1
	ASIL503	INTERMEDIATE JAPANESE I	X		1/1
	ASIL505	ADVANCED JAPANESE I	X		1/1
	ASIL507	BEGINNERS CHINESE I	X		1/1
	ASIL511	ADVANCED CHINESE I	X		1/1
ASIA (ASIAN STUDIES WITH THESIS)	U0 G16		U0 G3 I4	U0 G16 I6	U0 G19 I9
	ASIA509	COMPARATIVE POLITICAL ECONOMY IN ASIA		X	1/1
	ASIA593	DIRECTED READINGS IN ASIA STUDIES IV	X	X	2/2
	ASIA512	SOUTH ASIAN INTERNATIONAL REL.&POLITICS		X	1/1
	ASIA518	HISTORY OF MODERN JAPAN		X	1/1
	ASIA521	JAPANESE SOCIETY THROUGH ART,&CULTURE		X	1/1
	ASIA523	JAPANESE MANAGEMENT CULTURE&KNOW-HOW		X	1/1
	ASIA527	MODERN HISTORY OF CHINA		X	1/1
	ASIA536	SEMINAR ON MODERN KOREA		X	1/1
	ASIA579	GRADUATE SEMINAR	X	X	2/2
	ASIA591	DIRECTED READINGS IN ASIAN STUDIES II		X	1/1
	ASIA690	MASTER'S THESIS	X	X	2/2
	ASIL502	BEGINNER'S JAPANESE II		X	1/1
	ASIL504	INTERMEDIATE JAPANESE II		X	1/1