

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Choose an item.



Assignment Cover Sheet

Project Title:	Course Management system using file system in python		
Term :	Mid-term	Date of Submission:	8 November 2022
Course Title:	PROGRAMMING IN PYTHON [B]		
Course Code:	01333	Section:	B
Semester:	Fall	22-23	Course Teacher: AKINUL ISLAM JONY

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them material used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* Student(s) must complete all details except the faculty use part.

** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

No	Name	ID	Program	Signature
1	Rifat Hossain	20-42461-1	Bsc CSE	<i>Rifat</i>

Faculty use only

FACULTY COMMENTS	Marks Obtained	
	Total Marks	

Course Management system using file system in python

Project summary: In this project various technique were used to keep track of courses in local file system using python programming language. Operations like adding ,delating ,updating ,searching can be performed by this program.

Project requirements: There where several requirements such adding ,delating, updating , searching and displaying all the courses from file. To perform these task csv module is used . also prompt were shown if the necessary steps missing.

Add course: first course name, code ,pre-prequiside and credit is asked from user ,when necessary conditions were being met finally that course is added to the csv file .

Updating an existing Course: From user data following row in csv got updated.

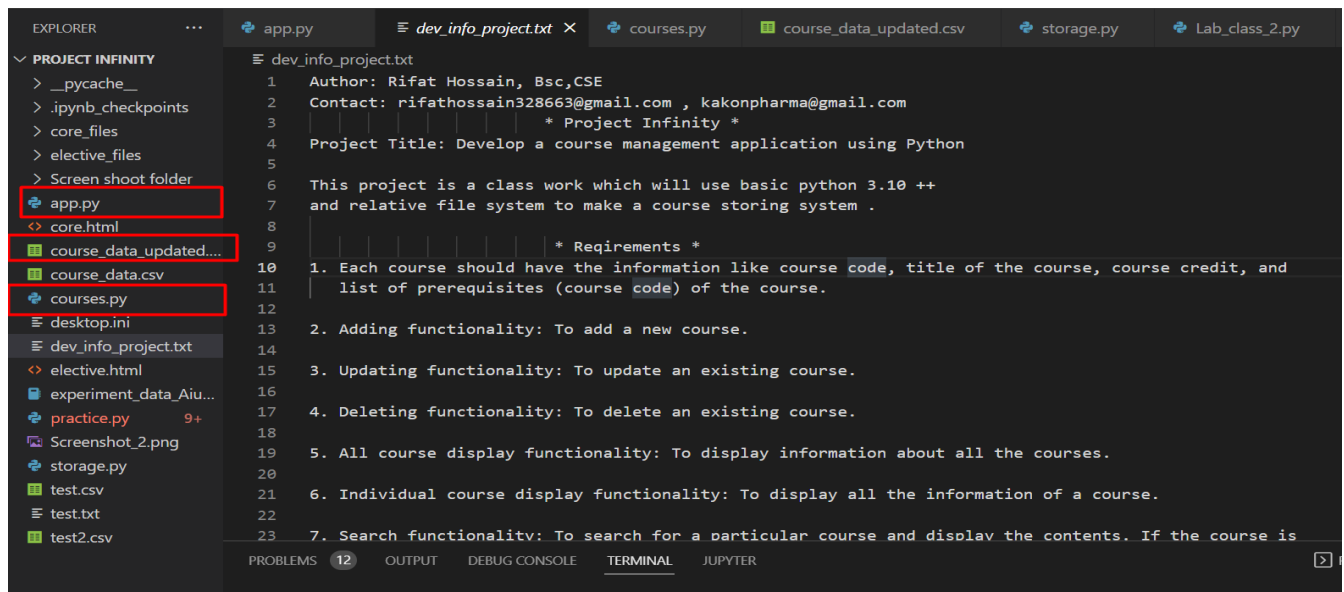
Delating an existing Course: Form user data delatting the course along with course code, index, and prerequisite.

Display courses: All the courses in specified csv file has been showed for the user.

Search Course: This option Shows a given coure exist or not, if that course exist then shows all the data related to that course.

Quit: exit out of the loop ,thus stoping the program.

Main Project files



[app.py](#)

Shows all the options and call object instances from Course class from Course.py

```
PS C:\Users\ASUS\Desktop\Project Infinity> & E:/python.exe "c:/Users/ASUS/Desktop/Project Infinity/app.py"
1: Add course

2: Update an existing course

3: Delete an existing course

4: Display information about all the courses

5: Search Course

6: quit

Select your input from these flowing options by pressing the SERIAL NUMBER
6 EX (1,2,3,4,5,6) FROM 1 TO 6 AND ENTER
```

Code:

```
app.py > ...
1  from csv import DictReader
2  from courses import Course
3
4
5  prompt = '\t Have a nice day'
6  user_input = ''
7  infos = ['1: Add course', '2: Update an existing course',
8           '3: Delete an existing course', '4: Display information about all the courses', '5: Search Course', '6:
           quit']
9  course_info = ['course_name', 'course_code',
10                'course_prerequizite_code', 'course_credit']
11 while True:
12     for info in infos:
13         print(f'{info}\n')
14     user_input = int(input(
15         'Select your input from these flowing options by pressing the SERIAL NUMBER \n6 EX (1,2,3,4,5,6) FROM 1 TO
        6 AND ENTER\n'))
16
17     if user_input == 1:
18         '''ADD A Course'''
19         courses = []
20         for course in course_info:
21             courses.append(input(f' \t Type {course} :'))
22 # if not empty check course exist or not
23     if courses[0:4] != '':
24         if Course.check(courses[0], courses[1]) == True:
25             print('\t this course already exist \n ')
26         elif Course.check2(courses[0], courses[1], courses[2]) == False:
```

```

app.py > ...
25         print('\t this course already exist \n')
26     elif Course.check2(courses[0], courses[1], courses[2]) == False:
27         # if course and pre req dosent exist then pass the value to constructor
28         course = Course(courses[0], courses[1],
29                           courses[2], int(courses[3]))
30         course.add_course()
31     # name ,code ,credit is mandatory
32     elif courses[0:2] == '' and courses[3] == '':
33         print('course name , code, credit is mandatory \n')
34         while True:
35             courses[0] = input('\t type course Name ? \n')
36             courses[1] = input('\t type course Code ? \n')
37             courses[3] = input('\t type course Credit ? \n')
38             if courses[0:2] != '' and courses[3] != '':
39                 # if name ,code ,credit is not empty then check course exist or not then pre
40
41                 if Course.check(courses[0], courses[1]) == True:
42                     print('\t this course already exist \n')
43                     break
44                 # course dosent exist and pre req dosent exist
45                 elif Course.check(courses[0], courses[1]) == False and courses[2] == '':
46                     answer = input(
47                         '\t would you like to enter a pre_requisite (y/n)\n')
48                     if answer == 'y':
49                         course[2] = input('\t enter pre requisite')
50                         course = Course(
51                             courses[0], courses[1], courses[2], int(courses[3]))
52                         course.add_course()
53                         break
54                     else:

```

```

54         else:
55             course = Course(
56                 courses[0], courses[1], courses[2], int(courses[3]))
57             course.add_course()
58             break
59
60     '''
61     course = Course(courses[0], courses[1], courses[2], int(courses[3]))
62     course.add_course()
63     '''
64
65     # -----
66     elif user_input == 2:
67         '''2: Update an existing course'''
68
69         courses = []
70         for course in course_info:
71             courses.append(input(f'Type {course} :'))
72
73         course = Course(courses[0], courses[1], courses[2], int(courses[3]))
74
75         preamiters = ['updated_code', 'updated_name',
76                       'updated_credit', 'updated_pre_requizite']
77         details = []
78         for paramiter in preamiters:
79             details.append(input(f'\tnew {paramiter} :'))
80         course.update_course(details[0], details[1],
81                               int(details[2]), details[3])
82
83     # -----

```

```

# -----
elif user_input == 3:
    '''3: Delete an existing course'''
    courses = []
    courses.append(input('Course Name: \t'))
    courses.append(input('Course code: \t'))
    courses.append(input('Course pre_requisite: \t'))
    courses.append(int(input('Course cratid: \t')))

    course = Course(courses[0], courses[1], courses[2], courses[3])
    course.delate_course(courses[1])

elif user_input == 4:
    Course.display_course('add')

elif user_input == 5:
    '''Search Course
    ...

    course_name = input('Enter course name : ')
    course_Code = input('Enter course code : ')

    Course.search_course(course_name, course_Code)

elif user_input == 6:
    break
else:
    print('please enter valid input')

print(prompt)

```

Courses.py

The file contain the Course class, contractor, methods , and decoder such as static method.

```

courses.py > Course > check2
1  from csv import writer
2  from csv import DictWriter, DictReader
3  import pandas as pd
4  from tempfile import NamedTemporaryFile
5  import shutil
6
7
8  class Course:
9      def __init__(self, course_name: str, course_code: str, course_prerequisite_code: str, course_credit: int):
10         assert course_credit >= 0, 'credit cant be negative number'
11
12         self.course_name = course_name
13         self.course_code = course_code
14         self.course_prerequisite_code = course_prerequisite_code
15         self.course_credit = course_credit
16
17     def add_course(self):
18         '''with open('course_data_updated.csv', 'w') as file:'''
19
20         with open('course_data_updated.csv', 'a') as file:
21             header = ('Index', 'Code', 'Name', 'Credit', 'Pre-Requisite')
22             index = 'Index'
23             key_course_code = 'Code'
24             key_course_name = 'Name'
25             key_course_credit = 'Credit'
26             key_course_prerequisite_code = 'Pre-Requisite'
27             csv_writer = DictWriter(
28                 file, fieldnames=header, lineterminator='\n')
29             df = pd.read_csv('course_data_updated.csv')
30             count = df['Index'].max()
31             count += 1
32             # csv_writer.writeheader()

```

```

33         csv_writer.writerow(
34             {
35                 index: count,
36                 key_course_code: self.course_code,
37                 key_course_name: self.course_name,
38                 key_course_credit: self.course_credit,
39                 key_course_prerequisite_code: self.course_prerequisite_code
40             }
41         )
42     }
43 )
44
45 def delate_course(self, del_course):
46     with open('course_data_updated.csv', 'r+') as file:
47         csv_reader = DictReader(file)
48         data = list(csv_reader)
49
50         flag = 0
51
52         with NamedTemporaryFile(mode='w', delete=False) as temp_file:
53             header = ('Index', 'Code', 'Name', 'Credit', 'Pre-Requisite')
54             csv_writer = DictWriter(
55                 temp_file, fieldnames=header, lineterminator='\n')
56             csv_writer.writeheader()
57             for row in data:
58                 if row['Code'] == del_course:
59                     flag = 1
60                     continue
61             csv_writer.writerow(row)

```

```

        if flag == 1:
            temp_path = temp_file.name # jamela indentation
            shutil.move(temp_file.name, 'course_data_updated.csv')
            print('your course got successfully deleted ')
        elif flag == 0:
            print('your course dosent exist ')

def display_course(self):
    with open('course_data_updated.csv', 'r') as file:

        csv_reader = DictReader(file, lineterminator='\n')
        for row in csv_reader:
            print(row)

def update_course(self, updated_code, updated_name, updated_credit, updated_pre_requizite):
    with open('course_data_updated.csv', 'r') as file:
        csv_reader = DictReader(file)
        data = list(csv_reader)

    with NamedTemporaryFile(mode='w', delete=False) as temp_file:
        header = ('Index', 'Code', 'Name', 'Credit', 'Pre-Requisite')

        csv_writer = DictWriter(
            temp_file, fieldnames=header, lineterminator='\n')
        csv_writer.writeheader()

        for row in data:
            if row['Name'] == self.course_name or row['Code'] == self.course_code:

```

```

        csv_writer.writeheader()

        for row in data:
            if row['Name'] == self.course_name or row['Code'] == self.course_code:

                row['Code'] = updated_code
                row['Name'] = updated_name
                row['Credit'] = updated_credit
                row['Pre-Requisite'] = updated_pre_requizite
            csv_writer.writerow(row)
        shutil.move(temp_file.name, 'course_data_updated.csv')

    @staticmethod
    def search_course(name, code):

        with open('course_data_updated.csv', 'r') as f:
            csv_reader = DictReader(f, lineterminator='\n')
            data = list(csv_reader)
            # print(data)
            for row in data:
                if row['Name'] == name or row['Code'] == code:
                    if row['Pre-Requisite'] == '':
                        nan = 'no Pre_Requisite'
                        print(
                            f'\n\t\t\tThis course exist in your Systrm \n \t Course Name: {row["Name"]} \n\t Course
                            Code: {row["Code"]} \n\t Course Credit: {row["Credit"]} \n\t {nan} \n')
                    else:
                        nan = row['Pre-Requisite']
                        print(
                            f'\n\t\t\tThis course exist in your Systrm \n \t Course Name: {row["Name"]} \n\t Course
                            Code: {row["Code"]} \n\t Course Credit: {row["Credit"]} \n\t {nan} \n')
                else:

```

```

                    Code: {row["Code"]} \n\t Course Credit: {row["Credit"]} \n\t {nan} \n')
            else:
                print(
                    '\t\t\tThis course dosent exist \n \t\t\t please enter option (1) to add this course.')

    @staticmethod
    def check(name, code):

        with open('course_data_updated.csv', 'r') as file:

            csv_reader = DictReader(file, lineterminator='\n')
            for row in csv_reader:
                if row['Name'] == name or row['Code'] == code:
                    return True
                    '''print('this Course Exist\n')'''
                else:
                    return False

    @staticmethod
    def check2(name, code, pre_req):

        with open('course_data_updated.csv', 'r') as file:

            csv_reader = DictReader(file, lineterminator='\n')
            for row in csv_reader:
                if row['Pre-Requisite'].split() == pre_req:
                    return True
                    '''print('this Course Exist\n')'''
                else:
                    return False

```

For Course Storage: `course_data_updated.csv` ,many of them were extracted from aiub portal using pandas module.

course_data_updated.csv

```
1  Index,Code,Name,Credit,Pre-Requisite
2  0,MAT1102,DIFFERENTIAL CALCULUS & CO-ORDINATE GEOMETRY,3 0 0 0 0,
3  1,PHY1101,PHYSICS 1,3 0 0 0 0,
4  2,PHY1102,PHYSICS 1 LAB,1 1 0 0 0,
5  3,ENG1101,ENGLISH READING SKILLS & PUBLIC SPEAKING,3 0 0 1 0,
6  4,CSC1102,INTRODUCTION TO PROGRAMMING,3 0 0 0 0,
7  5,CSC1103,INTRODUCTION TO PROGRAMMING LAB,1 0 1 0 0,
8  6,CSC1101,INTRODUCTION TO COMPUTER STUDIES,1 0 1 0 0,
9  7,CSC1204,DISCRETE MATHEMATICS,3 0 0 0 0,MAT1102  CSC1102
10 8,MAT1205,INTEGRAL CALCULUS & ORDINARY DIFFERENTIAL EQUATIONS,3 0 0 0 0,MAT1102
11 9,CSC1205,OBJECT ORIENTED PROGRAMMING 1,3 0 1 0 0,CSC1102  CSC1103
12 10,PHY1203,PHYSICS 2,3 0 0 0 0,PHY1101
13 11,PHY1204,PHYSICS 2 LAB,1 1 0 0 0,PHY1102
14 12,ENG1202,ENGLISH WRITING SKILLS & COMMUNICATION,3 0 0 1 0,ENG1101
15 13,COE2101,INTRODUCTION TO ELECTRICAL CIRCUITS,3 0 0 0 0,PHY1101
16 14,COE2102,INTRODUCTION TO ELECTRICAL CIRCUITS LAB,1 1 0 0 0,PHY1102
17 15,CHEM1101,CHEMISTRY,3 1 0 0 0,PHY1203
18 16,MAT2101,"COMPLEX VARIABLE,LAPLACE & Z-TRANSFORMATION",3 0 0 0 0,MAT1205
19 17,CSC2108,INTRODUCTION TO DATABASE,3 0 1 0 0,CSC1205
20 18,EEE2104,ELECTRONIC DEVICES LAB,1 1 0 0 0,COE2102
21 19,BBA1102,PRINCIPLES OF ACCOUNTING,3 0 0 0 0,MAT1205
22 20,EEE2103,ELECTRONIC DEVICES,3 0 0 0 0,COE2101
23 21,CSC2106,DATA STRUCTURE,3 0 0 0 0,CSC1204  CSC1205
24 22,CSC2107,DATA STRUCTURE LAB,1 0 1 0 0,CSC1204  CSC1205
25 23,BAE2101,COMPUTER AIDED DESIGN & DRAFTING,1 0 1 0 0,
26 24,CSC2211,ALGORITHMS,3 0 1 0 0,CSC2106  CSC2107
27 25,MAT2202,"MATRICES, VECTORS, FOURIER ANALYSIS",3 0 0 0 0,MAT2101
28 26,CSC2210,OBJECT ORIENTED PROGRAMMING 2,3 0 1 0 0,CSC2108  CSC1205
29 27,CSC2209,OBJECT ORIENTED ANALYSIS AND DESIGN,3 0 0 0 0,CSC2108
30 28,BAS2101,BANGLADESH STUDIES,3 0 0 0 0,CSC1101
31 29,EEE3101,DIGITAL LOGIC AND CIRCUITS,3 0 0 0 0,EEE2103
32 30,EEE3102,DIGITAL LOGIC AND CIRCUITS LAB,1 1 0 0 0,EEE2104

92 90,CSC4230,BIOINFORMATICS,3 0 0 0 0,CSC3217
93 91,CSC4231,PARALLEL COMPUTING,3 0 0 0 0,CSC2211  CSC3217
94 92,CSC4232,MACHINE LEARNING,3 0 0 0 0,CSC3217
95 93,COE4233,WIRELESS SENSOR NETWORKS,3 0 1 0 0,COE3206  COE3103
96 94,EEE4241,"INDUSTRIAL ELECTRONICS, DRIVES & INSTRUMENTATION",3 0 1 0 0,EEE3101
97 95,CSC4272,MOBILE APPLICATION DEVELOPMENT,3 0 1 0 0,CSC3215
98 96,CSC4273,SOFTWARE ARCHITECTURE AND DESIGN PATTERNS,3 0 0 0 0,CSC4160
99 97,MIS4007,DIGITAL MARKETING,3 0 0 0 0,MIS3101  CSC3215
100 98,MIS4012,"E-COMMERCE, E-GOVERNANCE & E-SERIES",3 0 0 0 0,CSC3215
101 99,EEE3103,DIGITAL SIGNAL PROCESSING,3 0 1 0 0,EEE2213
102 100,EEE4217,VLSI CIRCUIT DESIGN,3 0 1 0 0,EEE4241
103 101,s,s,4,f
104 102,hos,rifat,2,111
```


Operation Results:

Add course:

```
102 100,EEE4217,VLSI CIRCUIT DESIGN,3 0 1 0 0,EEE4241
103 101,s,s,4,f
104 102,hos,rifat,2,111
105

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

4: Display information about all the courses

5: Search Course

6: quit

Select your input from these flowing options by pressing the SERIAL NUMBER
6 EX (1,2,3,4,5,6) FROM 1 TO 6 AND ENTER
1
    Type course_name :rifat
    Type course_code :hos
    Type course_prerequizite_code :111
    Type course_credit :2
1: Add course

2: Update an existing course
```

Delete Course: (before delating course):

```
app.py courses.py course_data_updated.csv X storage.py Lab_class_2.py practice.py 9+
course_data_updated.csv
97 95,CSC4272,MOBILE APPLICATION DEVELOPMENT,3 0 1 0 0,CSC3215
98 96,CSC4273,SOFTWARE ARCHITECTURE AND DESIGN PATTERNS,3 0 0 0 0,CSC4160
99 97,MIS4007,DIGITAL MARKETING,3 0 0 0 0,MIS3101 CSC3215
100 98,MIS4012,"E-COMMERCE, E-GOVERNANCE & E-SERIES",3 0 0 0 0,CSC3215
101 99,EEE3103,DIGITAL SIGNAL PROCESSING,3 0 1 0 0,EEE2213
102 100,EEE4217,VLSI CIRCUIT DESIGN,3 0 1 0 0,EEE4241
103 101,a,b,3,e
```

(After deleting course:)

```

99 97,MIS4007,DIGITAL MARKETING,3 0 0 0 0,MIS3101 CSC3215
100 98,MIS4012,"E-COMMERCE, E-GOVERNANCE & E-SERIES",3 0 0 0 0,CSC3215
101 99,EEE3103,DIGITAL SIGNAL PROCESSING,3 0 1 0 0,EEE2213
102 100,EEE4217,VLSI CIRCUIT DESIGN,3 0 1 0 0,EEE4241
103

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

5: Search Course

6: quit

Select your input from these flowing options by pressing the SERIAL NUMBER

6 EX (1,2,3,4,5,6) FROM 1 TO 6 AND ENTER

3

Course Name: b

Course code: a

Course pre_requisite: e

Course cratid: 3

your course got successfully deleted

1: Add course

If that course does not exist (show prompt)

Select your input from these flowing options by pressing the SERIAL NUMBER

6 EX (1,2,3,4,5,6) FROM 1 TO 6 AND ENTER

3

Course Name: abu

Course code: bokkor

Course pre_requisite: 3

Course cratid: 3

your course dosent exist

1: Add course

Update Course:

```

101 99,EEE3103,DIGITAL SIGNAL PROCESSING,3 0 1 0 0,EEE2213
102 100,EEE4217,VLSI CIRCUIT DESIGN,3 0 1 0 0,EEE4241
103 101,s,s,4,f
104

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

6 EX (1,2,3,4,5,6) FROM 1 TO 6 AND ENTER

2

Type course_name :a

Type course_code :a

Type course_prerequizite_code :b

Type course_credit :3

new updated_code :s

new updated_name :s

new updated_credit :4

new updated_pre_requizite :f

1: Add course

2: Update an existing course

3: Delete an existing course

4: Display information about all the courses

5: Search Course

6: quit

```

4
{'Index': '0', 'Code': 'MAT1102', 'Name': 'DIFFERENTIAL CALCULUS & CO-ORDINATE GEOMETRY', 'Credit': '3 0 0 0', 'Pre-Requirement': ''}
{'Index': '1', 'Code': 'PHY1101', 'Name': 'PHYSICS 1', 'Credit': '3 0 0 0', 'Pre-Requirement': ''}
{'Index': '2', 'Code': 'PHY1102', 'Name': 'PHYSICS 1 LAB', 'Credit': '1 1 0 0', 'Pre-Requirement': ''}
{'Index': '3', 'Code': 'ENG1101', 'Name': 'ENGLISH READING SKILLS & PUBLIC SPEAKING', 'Credit': '3 0 0 1', 'Pre-Requirement': ''}
{'Index': '4', 'Code': 'CSC1102', 'Name': 'INTRODUCTION TO PROGRAMMING', 'Credit': '3 0 0 0', 'Pre-Requirement': ''}
{'Index': '5', 'Code': 'CSC1103', 'Name': 'INTRODUCTION TO PROGRAMMING LAB', 'Credit': '1 0 1 0', 'Pre-Requirement': ''}
{'Index': '6', 'Code': 'CSC1101', 'Name': 'INTRODUCTION TO COMPUTER STUDIES', 'Credit': '1 0 1 0', 'Pre-Requirement': ''}
{'Index': '7', 'Code': 'CSC1204', 'Name': 'DISCRETE MATHEMATICS', 'Credit': '3 0 0 0', 'Pre-Requirement': 'MAT1102 CSC1102'}
{'Index': '8', 'Code': 'MAT1205', 'Name': 'INTEGRAL CALCULUS & ORDINARY DIFFERENTIAL EQUATIONS', 'Credit': '3 0 0 0', 'Pre-Requirement': 'MAT1102'}
{'Index': '9', 'Code': 'CSC1205', 'Name': 'OBJECT ORIENTED PROGRAMMING 1', 'Credit': '3 0 1 0', 'Pre-Requirement': 'CSC1102 CSC1103'}
{'Index': '10', 'Code': 'PHY1203', 'Name': 'PHYSICS 2', 'Credit': '3 0 0 0', 'Pre-Requirement': 'PHY1101'}
{'Index': '11', 'Code': 'PHY1204', 'Name': 'PHYSICS 2 LAB', 'Credit': '1 1 0 0', 'Pre-Requirement': 'PHY1102'}
{'Index': '12', 'Code': 'ENG1202', 'Name': 'ENGLISH WRITING SKILLS & COMMUNICATION', 'Credit': '3 0 0 1', 'Pre-Requirement': 'ENG1101'}
{'Index': '13', 'Code': 'COE2101', 'Name': 'INTRODUCTION TO ELECTRICAL CIRCUITS', 'Credit': '3 0 0 0', 'Pre-Requirement': 'PHY1101'}
{'Index': '14', 'Code': 'COE2102', 'Name': 'INTRODUCTION TO ELECTRICAL CIRCUITS LAB', 'Credit': '1 1 0 0', 'Pre-Requirement': 'PHY1102'}
{'Index': '15', 'Code': 'CHEM1101', 'Name': 'CHEMISTRY', 'Credit': '3 1 0 0', 'Pre-Requirement': 'PHY1203'}
{'Index': '16', 'Code': 'MAT2101', 'Name': 'COMPLEX VARIABLE, LAPLACE & Z-TRANSFORMATION', 'Credit': '3 0 0 0', 'Pre-Requirement': 'MAT1205'}
{'Index': '17', 'Code': 'CSC2108', 'Name': 'INTRODUCTION TO DATABASE', 'Credit': '3 0 1 0', 'Pre-Requirement': 'CSC1205'}
{'Index': '18', 'Code': 'EEE2104', 'Name': 'ELECTRONIC DEVICES LAB', 'Credit': '1 1 0 0', 'Pre-Requirement': 'COE2102'}
{'Index': '19', 'Code': 'BBA1102', 'Name': 'PRINCIPLES OF ACCOUNTING', 'Credit': '3 0 0 0', 'Pre-Requirement': 'MAT1205'}
{'Index': '20', 'Code': 'EEE2103', 'Name': 'ELECTRONIC DEVICES', 'Credit': '3 0 0 0', 'Pre-Requirement': 'COE2101'}
{'Index': '21', 'Code': 'CSC2106', 'Name': 'DATA STRUCTURE', 'Credit': '3 0 0 0', 'Pre-Requirement': 'CSC1204 CSC1205'}
{'Index': '22', 'Code': 'CSC2107', 'Name': 'DATA STRUCTURE LAB', 'Credit': '1 0 1 0', 'Pre-Requirement': 'CSC1204 CSC1205'}
{'Index': '23', 'Code': 'BAE2101', 'Name': 'COMPUTER AIDED DESIGN & DRAFTING', 'Credit': '1 0 1 0', 'Pre-Requirement': ''}
{'Index': '24', 'Code': 'CSC2211', 'Name': 'ALGORITHMS', 'Credit': '3 0 1 0', 'Pre-Requirement': 'CSC2106 CSC2107'}
{'Index': '25', 'Code': 'MAT2202', 'Name': 'MATRICES, VECTORS, FOURIER ANALYSIS', 'Credit': '3 0 0 0', 'Pre-Requirement': 'MAT2101'}
{'Index': '26', 'Code': 'CSC2210', 'Name': 'OBJECT ORIENTED PROGRAMMING 2', 'Credit': '3 0 1 0', 'Pre-Requirement': 'CSC2108 CSC1205'}
{'Index': '27', 'Code': 'CSC2209', 'Name': 'OBJECT ORIENTED ANALYSIS AND DESIGN', 'Credit': '3 0 0 0', 'Pre-Requirement': 'CSC2108'}
{'Index': '28', 'Code': 'BAS2101', 'Name': 'BANGLADESH STUDIES', 'Credit': '3 0 0 0', 'Pre-Requirement': 'CSC1101'}
{'Index': '29', 'Code': 'EEE3101', 'Name': 'DIGITAL LOGIC AND CIRCUITS', 'Credit': '3 0 0 0', 'Pre-Requirement': 'EEE2103'}
{'Index': '30', 'Code': 'EEE3102', 'Name': 'DIGITAL LOGIC AND CIRCUITS LAB', 'Credit': '1 1 0 0', 'Pre-Requirement': 'EEE2104'}
{'Index': '31', 'Code': 'MAT3103', 'Name': 'COMPUTATIONAL STATISTICS AND PROBABILITY', 'Credit': '3 0 0 0', 'Pre-Requirement': 'MAT2101'}
{'Index': '32', 'Code': 'CSC3113', 'Name': 'THEORY OF COMPUTATION', 'Credit': '3 0 0 0', 'Pre-Requirement': 'CSC2211'}
{'Index': '33', 'Code': 'EC03150', 'Name': 'PRINCIPLES OF ECONOMICS', 'Credit': '2 0 0 0', 'Pre-Requirement': 'MAT3103'}
{'Index': '34', 'Code': 'ENG2103', 'Name': 'BUSINESS COMMUNICATION', 'Credit': '3 0 0 0', 'Pre-Requirement': 'BAS2101'}
{'Index': '35', 'Code': 'MAT3101', 'Name': 'NUMERICAL METHODS FOR SCIENCE AND ENGINEERING', 'Credit': '3 0 0 0', 'Pre-Requirement': 'MAT220

```

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python

6: quit

Select your input from these flowing options by pressing the SERIAL NUMBER
6 EX (1,2,3,4,5,6) FROM 1 TO 6 AND ENTER
5
Enter course name : PHYSICS 1
Enter course code : PHY1101

        This course exist in your Systrm
Course Name: PHYSICS 1
Course Code: PHY1101
Course Credit: 3 0 0 0 0
no Pre_Requisite

1: Add course

2: Update an existing course

```

```

In [23]: import pandas as pd
import html5lib
ur1='https://www.aiub.edu/faculties/fst/ug-course-catalog'
ur12='https://en.wikipedia.org/wiki/List_of_timelines#Environment'
html1='core.html'
html2='elective.html'

In [24]: list_of_df=pd.read_html(html1)

In [29]: list_of_df[3]

Out[29]:


|   | Code    | Name                                         | Credit(Lec-Sci-Comp-Lan-Stu) | Pre-Requisite |
|---|---------|----------------------------------------------|------------------------------|---------------|
| 0 | MAT1102 | DIFFERENTIAL CALCULUS & CO-ORDINATE GEOMETRY | 3 0 0 0                      | NaN           |
| 1 | PHY1101 | PHYSICS 1                                    | 3 0 0 0                      | NaN           |
| 2 | PHY1102 | PHYSICS 1 LAB                                | 1 1 0 0                      | NaN           |
| 3 | ENG1101 | ENGLISH READING SKILLS & PUBLIC SPEAKING     | 3 0 0 1                      | NaN           |
| 4 | CSC1102 | INTRODUCTION TO PROGRAMMING                  | 3 0 0 0                      | NaN           |
| 5 | CSC1103 | INTRODUCTION TO PROGRAMMING LAB              | 1 0 1 0                      | NaN           |
| 6 | CSC1101 | INTRODUCTION TO COMPUTER STUDIES             | 1 0 1 0                      | NaN           |



In [30]: list_of_df[4]

Out[30]:


|   | Code    | Name                                           | Credit(Lec-Sci-Comp-Lan-Stu) | Pre-Requisite   |
|---|---------|------------------------------------------------|------------------------------|-----------------|
| 0 | CSC1204 | DISCRETE MATHEMATICS                           | 3 0 0 0                      | MAT1102 CSC1102 |
| 4 | MAT1205 | INTEGRAL CALCULUS & ORDINARY DIFFERENTIAL EQUA | 3 0 0 0                      | MAT1102         |



# adding those 2 data frams to one final dataframes .csv file
#then application work

In [50]: #experiment on dataframs
df_all_rows = pd.concat([list_of_df[3],list_of_df[4],list_of_df[5],list_of_df[6],list_of_df[7],list_of_df[8],list_of_df[9],list_of_df[10],list_of_df[11],list_of_df[12],list_of_df[13],list_of_df[14],list_of_df[15],list_of_df[16],list_of_df[17],list_of_df[18],list_of_df[19],list_of_df[20],list_of_df[21],list_of_df[22],list_of_df[23],list_of_df[24],list_of_df[25],list_of_df[26],list_of_df[27],list_of_df[28],list_of_df[29],list_of_df[30],list_of_df[31],list_of_df[32],list_of_df[33],list_of_df[34],list_of_df[35],list_of_df[36],list_of_df[37],list_of_df[38],list_of_df[39],list_of_df[40],list_of_df[41],list_of_df[42],list_of_df[43],list_of_df[44],list_of_df[45],list_of_df[46],list_of_df[47],list_of_df[48],list_of_df[49],list_of_df[50],list_of_df[51],list_of_df[52],list_of_df[53],list_of_df[54],list_of_df[55],list_of_df[56],list_of_df[57],list_of_df[58],list_of_df[59],list_of_df[60],list_of_df[61],list_of_df[62],list_of_df[63],list_of_df[64],list_of_df[65],list_of_df[66],list_of_df[67],list_of_df[68],list_of_df[69],list_of_df[70],list_of_df[71],list_of_df[72],list_of_df[73],list_of_df[74],list_of_df[75],list_of_df[76],list_of_df[77],list_of_df[78],list_of_df[79],list_of_df[80],list_of_df[81],list_of_df[82],list_of_df[83],list_of_df[84],list_of_df[85],list_of_df[86],list_of_df[87],list_of_df[88],list_of_df[89],list_of_df[90],list_of_df[91],list_of_df[92],list_of_df[93],list_of_df[94],list_of_df[95],list_of_df[96],list_of_df[97],list_of_df[98],list_of_df[99],list_of_df[100],list_of_df[101],list_of_df[102],list_of_df[103],list_of_df[104],list_of_df[105],list_of_df[106],list_of_df[107],list_of_df[108],list_of_df[109],list_of_df[110],list_of_df[111],list_of_df[112],list_of_df[113],list_of_df[114],list_of_df[115],list_of_df[116],list_of_df[117],list_of_df[118],list_of_df[119],list_of_df[120],list_of_df[121],list_of_df[122],list_of_df[123],list_of_df[124],list_of_df[125],list_of_df[126],list_of_df[127],list_of_df[128],list_of_df[129],list_of_df[130],list_of_df[131],list_of_df[132],list_of_df[133],list_of_df[134],list_of_df[135],list_of_df[136],list_of_df[137],list_of_df[138],list_of_df[139],list_of_df[140],list_of_df[141],list_of_df[142],list_of_df[143],list_of_df[144],list_of_df[145],list_of_df[146],list_of_df[147],list_of_df[148],list_of_df[149],list_of_df[150],list_of_df[151],list_of_df[152],list_of_df[153],list_of_df[154],list_of_df[155],list_of_df[156],list_of_df[157],list_of_df[158],list_of_df[159],list_of_df[160],list_of_df[161],list_of_df[162],list_of_df[163],list_of_df[164],list_of_df[165],list_of_df[166],list_of_df[167],list_of_df[168],list_of_df[169],list_of_df[170],list_of_df[171],list_of_df[172],list_of_df[173],list_of_df[174],list_of_df[175],list_of_df[176],list_of_df[177],list_of_df[178],list_of_df[179],list_of_df[180],list_of_df[181],list_of_df[182],list_of_df[183],list_of_df[184],list_of_df[185],list_of_df[186],list_of_df[187],list_of_df[188],list_of_df[189],list_of_df[190],list_of_df[191],list_of_df[192],list_of_df[193],list_of_df[194],list_of_df[195],list_of_df[196],list_of_df[197],list_of_df[198],list_of_df[199],list_of_df[200],list_of_df[201],list_of_df[202],list_of_df[203],list_of_df[204],list_of_df[205],list_of_df[206],list_of_df[207],list_of_df[208],list_of_df[209],list_of_df[210],list_of_df[211],list_of_df[212],list_of_df[213],list_of_df[214],list_of_df[215],list_of_df[216],list_of_df[217],list_of_df[218],list_of_df[219],list_of_df[220],list_of_df[221],list_of_df[222],list_of_df[223],list_of_df[224],list_of_df[225],list_of_df[226],list_of_df[227],list_of_df[228],list_of_df[229],list_of_df[230],list_of_df[231],list_of_df[232],list_of_df[233],list_of_df[234],list_of_df[235],list_of_df[236],list_of_df[237],list_of_df[238],list_of_df[239],list_of_df[240],list_of_df[241],list_of_df[242],list_of_df[243],list_of_df[244],list_of_df[245],list_of_df[246],list_of_df[247],list_of_df[248],list_of_df[249],list_of_df[250],list_of_df[251],list_of_df[252],list_of_df[253],list_of_df[254],list_of_df[255],list_of_df[256],list_of_df[257],list_of_df[258],list_of_df[259],list_of_df[260],list_of_df[261],list_of_df[262],list_of_df[263],list_of_df[264],list_of_df[265],list_of_df[266],list_of_df[267],list_of_df[268],list_of_df[269],list_of_df[270],list_of_df[271],list_of_df[272],list_of_df[273],list_of_df[274],list_of_df[275],list_of_df[276],list_of_df[277],list_of_df[278],list_of_df[279],list_of_df[280],list_of_df[281],list_of_df[282],list_of_df[283],list_of_df[284],list_of_df[285],list_of_df[286],list_of_df[287],list_of_df[288],list_of_df[289],list_of_df[290],list_of_df[291],list_of_df[292],list_of_df[293],list_of_df[294],list_of_df[295],list_of_df[296],list_of_df[297],list_of_df[298],list_of_df[299],list_of_df[300],list_of_df[301],list_of_df[302],list_of_df[303],list_of_df[304],list_of_df[305],list_of_df[306],list_of_df[307],list_of_df[308],list_of_df[309],list_of_df[310],list_of_df[311],list_of_df[312],list_of_df[313],list_of_df[314],list_of_df[315],list_of_df[316],list_of_df[317],list_of_df[318],list_of_df[319],list_of_df[320],list_of_df[321],list_of_df[322],list_of_df[323],list_of_df[324],list_of_df[325],list_of_df[326],list_of_df[327],list_of_df[328],list_of_df[329],list_of_df[330],list_of_df[331],list_of_df[332],list_of_df[333],list_of_df[334],list_of_df[335],list_of_df[336],list_of_df[337],list_of_df[338],list_of_df[339],list_of_df[340],list_of_df[341],list_of_df[342],list_of_df[343],list_of_df[344],list_of_df[345],list_of_df[346],list_of_df[347],list_of_df[348],list_of_df[349],list_of_df[350],list_of_df[351],list_of_df[352],list_of_df[353],list_of_df[354],list_of_df[355],list_of_df[356],list_of_df[357],list_of_df[358],list_of_df[359],list_of_df[360],list_of_df[361],list_of_df[362],list_of_df[363],list_of_df[364],list_of_df[365],list_of_df[366],list_of_df[367],list_of_df[368],list_of_df[369],list_of_df[370],list_of_df[371],list_of_df[372],list_of_df[373],list_of_df[374],list_of_df[375],list_of_df[376],list_of_df[377],list_of_df[378],list_of_df[379],list_of_df[380],list_of_df[381],list_of_df[382],list_of_df[383],list_of_df[384],list_of_df[385],list_of_df[386],list_of_df[387],list_of_df[388],list_of_df[389],list_of_df[390],list_of_df[391],list_of_df[392],list_of_df[393],list_of_df[394],list_of_df[395],list_of_df[396],list_of_df[397],list_of_df[398],list_of_df[399],list_of_df[400],list_of_df[401],list_of_df[402],list_of_df[403],list_of_df[404],list_of_df[405],list_of_df[406],list_of_df[407],list_of_df[408],list_of_df[409],list_of_df[410],list_of_df[411],list_of_df[412],list_of_df[413],list_of_df[414],list_of_df[415],list_of_df[416],list_of_df[417],list_of_df[418],list_of_df[419],list_of_df[420],list_of_df[421],list_of_df[422],list_of_df[423],list_of_df[424],list_of
```