

Department of Computer Science and Engineering

Laboratory Manual

of

IT3008 - Programming with Python

is submitted to

Mr. Vishvajit Bakrola

Assistant Professor



Asha M. Tarsadia Institute of Computer Science and Technology

Uka Tarsadia University, Maliba Campus Bardoli, Gujarat

Semester – 2

(Summer 2022)



ASHA M. TARSADIA INSTITUTE OF COMPUTER SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that

Mr/Ms. _____, Enrollment

No: _____ of B.Tech. Computer Science and Engineering 2nd semester has satisfactorily completed his/her laboratory work of **IT3008 - Programming with Python** during regular term in academic year 2021-22.

Date of Submission: _____

Mr. Vishvajit Bakrola

Subject Teacher

Assistant Professor

Department of Computer Science and
Engineering

Department Authority

Institute Stamp

IT3008 - Programming with Python

Laboratory Practical Index

Sr. No.	Name of Practical	Actual Date
1	Write a python program to greet users with welcome messages using print() method.	21/02/2022
2	<p>Write a python program to demonstrate the creation of List data structure along with its methods - append(), extend(), insert(), remove(), clear(), index(), count(), sort(), reverse(), and copy().</p> <ul style="list-style-type: none"> I. Demonstrate positive and negative indexing with python List. II. Demonstrate slicing operations on python List. III. Demonstrate updation on List elements in python. IV. Demonstrate deletion of a single python list element and multiple elements using slicing operator. 	21/02/2022
3	<p>Write a python program to demonstrate the creation of tuples along with its methods - count() and index().</p> <ul style="list-style-type: none"> I. Demonstrate positive and negative indexing with python Tuple. II. Demonstrate slicing operations on python Tuple. 	28/02/2022
4	<p>Write a python program to demonstrate the creation of a Dictionary <i>student</i> with the <i>name</i>, <i>age</i> and <i>branch</i> of a student.</p> <ul style="list-style-type: none"> I. Demonstrate the updation of python dictionary. II. Demonstrate the removal of elements from the python dictionary. III. Demonstrate the use of following dictionary methods - clear(), copy(), get(), items(), keys(), popitem(), and values(). 	28/02/2022
5	Demonstrate the use of basic string methods in python - lower(), upper(), join(), split(), find(), and replace().	07/03/2022
6	Write a python program to implement basic arithmetic operations on user entered numbers.	07/03/2022
7	Write a python program to count how many times a specific number is occurring in a list. Take user input for both numbers and a list.	07/03/2022
8	Write a python program that takes N number of integers from the user in a	14/03/2022

	python list. Create a function that takes the list of user entered numbers and returns MAX and MIN numbers from that list to the user.	
9	Write a python program to perform basic matrix operations on user entered matrices.	14/03/2022
10	Write a python program to perform basic operations of a calculator. Provide choice for operations to users and make a program iterative. Provide specific exit option to users.	21/03/2022
11	Write a python program to demonstrate the use of arbitrary arguments.	21/03/2022
12	Create a class named <i>student</i> having attributes - <i>std_name</i> , <i>std_age</i> , <i>std_branch</i> , and <i>std_city</i> . Create a method named <i>get_data()</i> in <i>student</i> class that takes user input for these attributes and a method named <i>display()</i> that prints the attribute values on the terminal. Call both the methods by creating an instance <i>std_obj</i> of the class <i>student</i> .	21/03/2022
13	Write a python program to demonstrate basic banking operations. Create a class named <i>banking</i> having separate class methods for each operation. Call each method with an instance of the class and attribute values to be taken from the user.	28/03/2022
14	Create a class named <i>employee</i> having attributes - <i>emp_name</i> , <i>emp_age</i> , and <i>emp_city</i> . Create a method named <i>get_data()</i> in <i>employee</i> class that takes user input for these attributes. Derive a class named <i>emp_derived()</i> from the <i>employee</i> class, having an <i>__init__()</i> method that displays the attributes of the <i>employee</i> class upon instantiation.	28/03/2022
15	<p>Create a base class named <i>university</i> with its attributes - <i>name</i>, <i>year_of_estd</i>, and <i>city</i>. Derive following class from the super class <i>university</i>: <i>professor</i>, <i>lab_assistant</i>, <i>office_assistant</i>, and <i>peon</i>. Make the program choice based on the user. The attributes and method of various class are as below:</p> <ul style="list-style-type: none"> - Attributes of <i>professor</i> class: <i>designation</i>, <i>highest_qualification</i>, <i>area_of_research</i>, <i>year_of_joining</i>, <i>year_of_experience</i>, and <i>name_of_institute</i>. - Methods of professor class: <i>__init__()</i> method that gets invoked upon instantiation and takes values of class attributes. The <i>display()</i> method that prints class attribute values along with attributes of its super class. - Attributes of <i>lab_assistant</i> class: <i>designation</i> = “Lab Assistant” (static), <i>highest_qualification</i>, <i>additiobnal_skilss</i>, <i>year_of_joining</i>, and <i>name_of_institue</i>. 	04/04/2022

	<ul style="list-style-type: none"> - Methods of <i>lab_assistant</i> class: <code>__init__()</code> method that gets invoked upon instantiation and takes values of class attributes. The <code>display()</code> method that prints class attribute values along with attributes of its super class. - Attributes of <i>office_assistant</i> class: <code>designation</code> = “Office Assistant” (static), <code>highest_qualification</code>, <code>year_of_joining</code>, and <code>name_of_institute</code>. - Methods of <i>office_assistant</i> class: <code>__init__()</code> method that gets invoked upon instantiation and takes values of class attributes. The <code>display()</code> method that prints class attribute values along with attributes of its super class. - Attributes of <i>peon</i> class: <code>job_role</code> = “office Peon” (static), <code>qualification</code>, <code>year_of_joining</code>, and <code>name_of_institute</code>. - Methods of <i>peon</i> class: <code>__init__()</code> method that gets invoked upon instantiation and takes values of class attributes. The <code>display()</code> method that prints class attribute values along with attributes of its super class. 	
16	<p>Create three classes named - <i>C</i>, <i>Python</i>, and <i>Web_Designing</i> each having two primary attributes as <i>learnings_</i> and <i>name_of_professor</i>. Derive a class named <i>student</i> from these classes. The <i>student</i> class has following methods and attributes:</p> <ol style="list-style-type: none"> I. Global <code>std_college</code> attribute with static values. II. <code>__init__()</code> method with attributes - <code>std_name</code>, <code>std_enrollment_no</code>, and <code>std_course</code>. III. <code>display()</code> method to display various attribute values of the terminal. 	04/04/2022
17	Write a python program to demonstrate the use of data hiding.	18/04/2022
18	Write a python program to create a class named <i>area</i> . Define a class method <code>find_area()</code> that can find areas of different shapes whose value is given by the user. Invoke the class method by instantiation and prove method overloading.	18/04/2022
19	Write a python program to demonstrate the use of method overriding.	18/04/2022
20	Write a python program to demonstrate the use of try-catch block for exception handling.	18/04/2022
21	Write a python program to raise an exception with the python <code>raise</code> keyword.	25/04/2022
22	Write a python program to demonstrate the try-finally block.	25/04/2022
23	Write a python program to read the content of a file and return the number of words in a file to the user.	25/04/2022

24	Write a python program to read and show first N lines to the user. The number of lines N will be taken from user input.	02/05/2022
25	Write a python program that takes input of a student course from the user and and write it in a file.	02/05/2022
26	Write a python program to copy the content of one file to another file.	09/05/2022
27	Write a python program that creates 26 text files named A.txt, B.txt, and up to Z.txt.	09/05/2022

Practical 1

**Write a python program to greet users with welcome messages using
print() method.**

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 print("Hello ,Welcome to python programing")
```

Activities Visual Studio Code

Thu 20:14 •

program1.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program1.py"
Hello ,Welcome to python programing
202103103510429@utu-cgpit:~$
```

Ln 3, Col 22 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 2

Write a python program to demonstrate the creation of List data structure along with its methods - append(), extend(), insert(), remove(), clear(), index(), count(), sort(), reverse(), and copy().

- a. Demonstrate positive and negative indexing with python List.**
- b. Demonstrate slicing operations on python List.**
- c. Demonstrate updation on List elements in python.**
- d. Demonstrate deletion of a single python list element and multiple elements using slicing operator..**

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 list=[1,2,3,4,5,"Purv Joshi"]
6 list.append(5)
7 print(list)
8
9 l=[6,7,9,8,10]
10 list.extend(l)
11 print(list)
12
13 list.insert(1,"apple")
14 print(list)
15
16 list.remove("Joshi")
17 print(list)
18
19 print(list.index(2))
20
21 list.clear()
22 print(list)
23
24 print(list.count(2))
25
26 l.sort()
27 print(l)
28
29 l.reverse()
30 print(l)
31
32 x=l.copy()
33 print(x)
34
35 lst=[7,8,9,3,4,5]
36 print(lst[3])
37 print(lst[-2])
38 print(lst[2:4])
```

Activities Visual Studio Code Thu 20:15 •

program2.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program2.py"
[1, 2, 3, 4, 5, 'Purv Joshi', 5]
[1, 2, 3, 4, 5, 'Purv Joshi', 5, 6, 7, 9, 8, 10]
[1, 'apple', 2, 3, 4, 5, 'Purv Joshi', 5, 6, 7, 9, 8, 10]
Traceback (most recent call last):
  File "/home/202103103510429/Desktop/program/program2.py", line 16, in <module>
    list.remove("Joshi")
ValueError: list.remove(x): x not in list
202103103510429@utu-cgpit:~$
```

Ln 30, Col 9 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 3

Write a python program to demonstrate the creation of tuples along with its methods - count() and index().

- a. Demonstrate positive and negative indexing with python Tuple.**
- b. Demonstrate slicing operations on python Tuple.**
- c. Demonstrate updation on Tuple elements in python.**

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 tuple=(1,3,5,-2,1)
6
7 print(tuple.count(1))
8
9 print(tuple.index(-2))
10
11 print(tuple[1:4])
```

Activities Visual Studio Code

Thu 20:16 •

program3.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program3.py"
2
3
(3, 5, -2)
202103103510429@utu-cgpit:~$
```

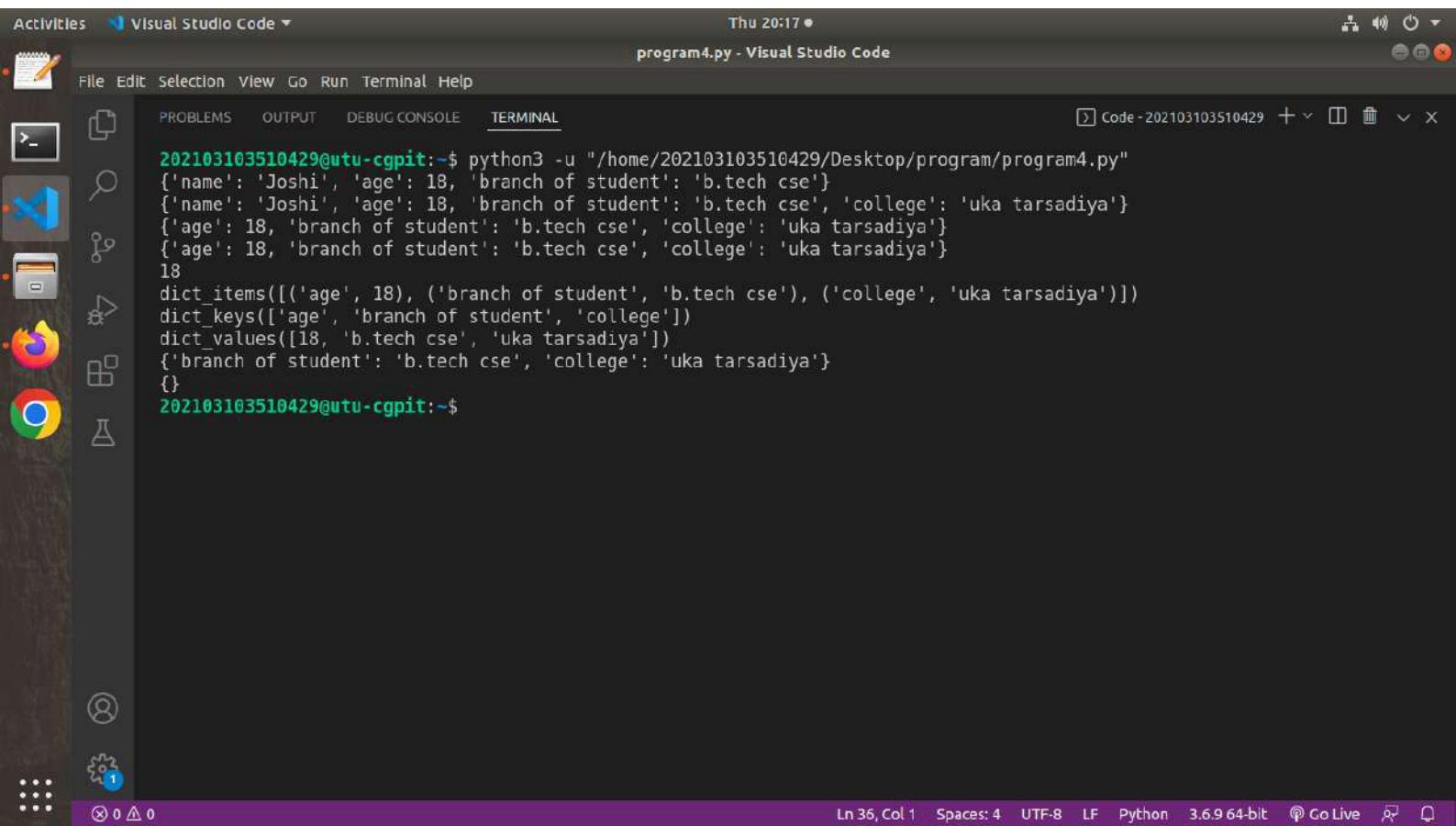
Ln 11, Col 18 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 4

Write a python program to demonstrate the creation of a Dictionary student with the name, age and branch of a student.

- a. Demonstrate the updation of python dictionary.**
- b. Demonstrate the removal of elements from the python dictionary.**
- c. Demonstrate the use of following dictionary methods - clear(), copy(), get(), items(), keys(), popitem(), and values().**

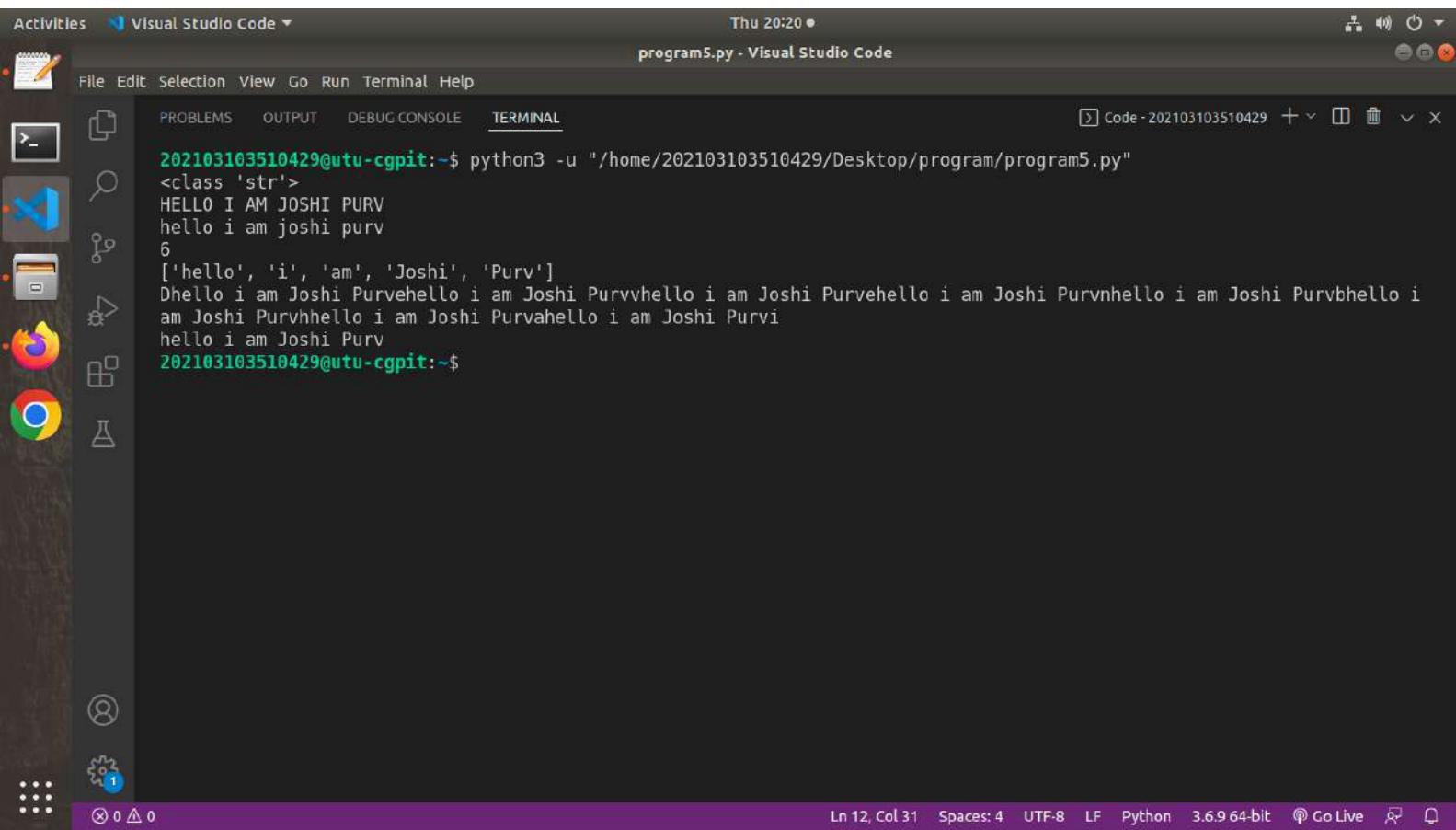
```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 dict={
6     "name":"Joshi",
7     "age":18,
8     "branch of student":"b.tech cse"
9 }
10 updatedict={
11     "college":"uka tarsadiya"
12 }
13 print(dict)
14 dict.update(updatedict)
15 print(dict)
16
17 dict.pop("name")
18 print(dict)
19
20 dict.copy()
21 print(dict)
22
23 print(dict.get("age"))
24
25 print(dict.items())
26
27 print(dict.keys())
28
29 print(dict.values())
30
31 dict.pop("age")
32 print(dict)
33
34 dict.clear()
35 print(dict)
```



Practical 5

**Demonstrate the use of basic string methods in python - lower(), upper(),
join(), split(), find(), and replace().**

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 r="hello i am Joshi Purv"
6 print(type(r))
7 print(r.upper())
8 print(r.lower())
9 print(r.find("i"))
10 print(r.split())
11 print(r.join("Devenbhai"))
12 print(r.replace('purv','car'))
```



Practical 6

Write a python program to implement basic arithmetic operations on user entered numbers.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenhai Joshi
4
5 a=int(input("enter the number 1= "))
6 b=int(input("enter the number 2= "))
7
8 print("the Add= ",a+b)
9 print("the Sub= ",a-b)
10 print("the Mul= ",a*b)
11 print("the Div= ",a/b)
```

Activities Visual Studio Code

Thu 20:21 •

program6.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program6.py"
enter the number 1= 5
enter the number 2= 10
the Add= 15
the Sub= -5
the Mul= 50
the Div= 0.5
202103103510429@utu-cgpit:~$
```

Ln 11, Col 23 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 7

Write a python program to count how many times a specific number is occurring in a list. Take user input for both numbers and a list.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenhai Joshi
4
5 def counting(com):
6     for i in range(len(com)):
7         n=com.count(com[i])
8
9
10    print("The element ",com[i],"repeats",n,"times")
11
12
13
14 l=[ ]
15 m=int(input("Enter the number of element in list 1: "))
16 for i in range(m):
17     l.append(int(input("enter the value = ")))
18 print(l)
19
20 lst=[]
21 n=int(input("Enter the number of element in list 2: "))
22 for i in range(n):
23     lst.append(int(input("enter the value = ")))
24 print(lst)
25
26 com=l+lst
27 print(com)
28 print(counting(com))
```

Activities Visual Studio Code Thu 20:24 •

program7.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program7.py"
Enter the number of element in list 1: 1
enter the value = 234
[234]
Enter the number of element in list 2: 2
enter the value = 3
enter the value = 45467
[3, 45467]
[234, 3, 45467]
The element 234 repeats 1 times
The element 3 repeats 1 times
The element 45467 repeats 1 times
None
202103103510429@utu-cgpit:~$
```

Ln 9, Col 1 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 8

Write a python program that takes N number of integers from the user in a python list. Create a function that takes the list of user entered numbers and returns MAX and MIN numbers from that list to the user.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 lst = []
6 while(1):
7     num = input("Enter number (enter exit to quit): ")
8     if num=="exit": break;
9     try:
10         lst.append(int(num))
11     except:
12         print("Enter a valid number")
13
14
15 def func(*lst):
16     large = lst[0]
17     small = lst[0]
18     for i in range(len(lst)):
19         if large < lst[i]: large = lst[i]
20         if small > lst[i]: small = lst[i]
21     return "Largest Number: " + str(large) + "\nSmallest Number: " + str(small)
22
23
24 print(func(*lst))
```

Activities Visual Studio Code

Thu 20:29 •

program8.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program8.py"
Enter number (enter exit to quit): 12
Enter number (enter exit to quit): -5
Enter number (enter exit to quit): 34
Enter number (enter exit to quit): -6
Enter number (enter exit to quit): exit
Largest Number: 34
Smallest Number: -6
202103103510429@utu-cgpit:~$
```

Ln 3, Col 22 Tab Size: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 9

Write a python program to perform basic matrix operations on user entered matrices.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 def add(matrix_1,matrix_2):
6     output=[]
7     for index_1 in range(len(matrix_1)):
8         r=[]
9         for index_2 in range(len(matrix_1[0])):
10            r.append(matrix_1[index_1][index_2]+matrix_2[index_1][index_2])
11        output.append(r)
12    return output
13
14 def sub(matrix_1,matrix_2):
15     output=[]
16     for index_1 in range(len(matrix_1)):
17         r=[]
18         for index_2 in range(len(matrix_1[0])):
19            r.append(matrix_1[index_1][index_2]-matrix_2[index_1][index_2])
20        output.append(r)
21    return output
22
23 def transpose(matrix):
24     transpose_matrix=[]
25     for index_1 in range(len(matrix)):
26         output=[]
27         for index_2 in range(len(matrix[0])):
28             output.append(matrix[index_2][index_1])
29         transpose_matrix.append(output)
30     return transpose_matrix
31
32 print(''1 For Addtion
33 2 For Subtraction
34 3 For Transpose
35 ''')
36 choice=int(input("Enter the choice = "))
37 print("matrix1 and matrix2 is equal to there row and column !!\n")
38 if choice==1:
39     matrix_1=[]
40     l1=[]
41     l2=[]
42     print("For matrix 1:-")
43     row=int(input("Enter the numer of row:"))
44     column=int(input("Enter the number of column:"))
45     for i in range(row):
46         l1.append(int(input("Enter value :")))
47     for j in range(column):
48         l2.append(int(input("Enter value :")))
49
50     matrix_1.append(l1)
51     matrix_1.append(l2)
52     print(matrix_1)
53
54     matrix_2=[]
55     l4=[]
56     l5=[]
57     print("For matrix 2:")
```

```
58 row=int(input("Enter the number of row:"))
59 column=int(input("Enter the number of column: "))
60 for i in range(row):
61     14.append(int(input("Enter value :")))
62 for j in range(column):
63     15.append(int(input("Enter value :")))
64
65 matrix_2.append(14)
66 matrix_2.append(15)
67 print(matrix_2)
68 print("Addtion of ",add(matrix_1, matrix_2))
69
70
71 if choice==2:
72     matrix_1=[]
73     11=[]
74     12=[]
75     print("For matrix 1:")
76     row=int(input("Enter the number of row:"))
77     column=int(input("Enter the number of column: "))
78     for i in range(row):
79         11.append(int(input("Enter value :")))
80     for j in range(column):
81         12.append(int(input("Enter value :")))
82
83 matrix_1.append(11)
84 matrix_1.append(12)
85 print(matrix_1)
86
87 matrix_2=[]
88 14=[]
89 15=[]
90 print("For matrix 2:")
91 row=int(input("Enter the number of row:"))
92 column=int(input("Enter the number of column: "))
93 for i in range(row):
94     14.append(int(input("Enter value :")))
95 for j in range(column):
96     15.append(int(input("Enter value :")))
97
98 matrix_2.append(14)
99 matrix_2.append(15)
100 print(matrix_2)
101 print("Subtraction of ",sub(matrix_1, matrix_2))
102
103 if choice==3:
104     matrix=[]
105     11=[]
106     12=[]
107     print("For matrix:")
108     row=int(input("Enter the number of row:"))
109     column=int(input("Enter the number of column: "))
110     for i in range(row):
111         11.append(int(input("Enter value :")))
112     for j in range(column):
113         12.append(int(input("Enter value :")))
114
```

```
115     matrix.append(l1)
116     matrix.append(l2)
117     print(matrix)
118     print("Transpore of ",transpore(matrix))
```

Activities Visual Studio Code • Thu 20:31 •

program9.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
20210310351042@mi-Opti:~$ python3 -u "/home/20210310351042/Desktop/program/program9.py"
1 For Addition
2 For Subtraction
3 For Transpose

Enter the choice = 1
matrix1 and matrix2 is equal to there row and column !!

For matrix 1:-
Enter the number of row:2
Enter the number of column:2
Enter value :1
Enter value :2
Enter value :3
Enter value :4
[[1, 2], [3, 4]]
For matrix 2:-
Enter the number of row:2
Enter the number of column: 2
Enter value :-5
Enter value :-6
Enter value :2
Enter value :0
[[-5, -6], [2, 0]]
Addition of [[1, 4], [-4], [30, 10]]
20210310351042@mi-Opti:~$ python3 -u "/home/20210310351042/Desktop/program/program9.py"
1 For Addition
2 For Subtraction
3 For Transpose

Enter the choice = 2
matrix1 and matrix2 is equal to there row and column !!

For matrix 1:-
Enter the number of row:2
Enter the number of column: 2
Enter value :24
Enter value :54
Enter value :3
Enter value :8
[[24, 54], [3, 8]]
For matrix 2:-
Enter the number of row:2
Enter the number of column: 2
Enter value :23
Enter value :4
Enter value :45
Enter value :6
[[23, 4], [45, 6]]
Subtraction of [[21, 52], [-35, 21]]
20210310351042@mi-Opti:~$
```

Ln 73, Col 10 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live ⌂

Practical 10

**Write a python program to perform basic operations of a calculator.
Provide choice for operations to users and make a program iterative.
Provide specific exit option to users.**

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 def sum(list):
6     a=0
7     for i in range(0, len(list)):
8         a=a+list[i]
9     return a
10 def mul(list):
11     a=1
12     for i in range(len(list)):
13         a=a*list[i]
14     return a
15
16 def div(n,n1):
17     return n/n1
18
19 def sub(list):
20     a=1
21     for i in range(len(list)):
22         a=a-list[i]
23     return a
24
25
26 def oe(n):
27     if n%2==0:
28         print('even number')
29     else:
30         print('odd number')
31
32
33 ask=input('''1.Addtion
34 2.Multiple
35 3.Divide
36 4.Odd or Even
37 5.Subtraction
38 choice your number= ''')
39 if ask=='1':
40     n=int(input('enter the value='))
41     list=[]
42     for i in range(0,n):
43         list.append(int(input()))
44     print(sum(list))
45
46 elif ask=='2':
47     n=int(input('enter the value='))
48     list=[]
49     for i in range(n):
50         list.append(int(input()))
51     print(mul(list))
52
53 elif ask=='3':
54     n=int(input('enter the value='))
55     n1=int(input('enter the value='))
56     print(div(n,n1))
57
```

```
58 elif ask=='5':  
59     n=int(input('enter the value='))  
60     list=[]  
61     for i in range(n):  
62         list.append(int(input()))  
63     print(sub(list))  
64  
65 elif ask=='4':  
66     n=(int(input()))  
67     print(oe(n))
```

Activities Visual Studio Code Thu 20:34 •

program10.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

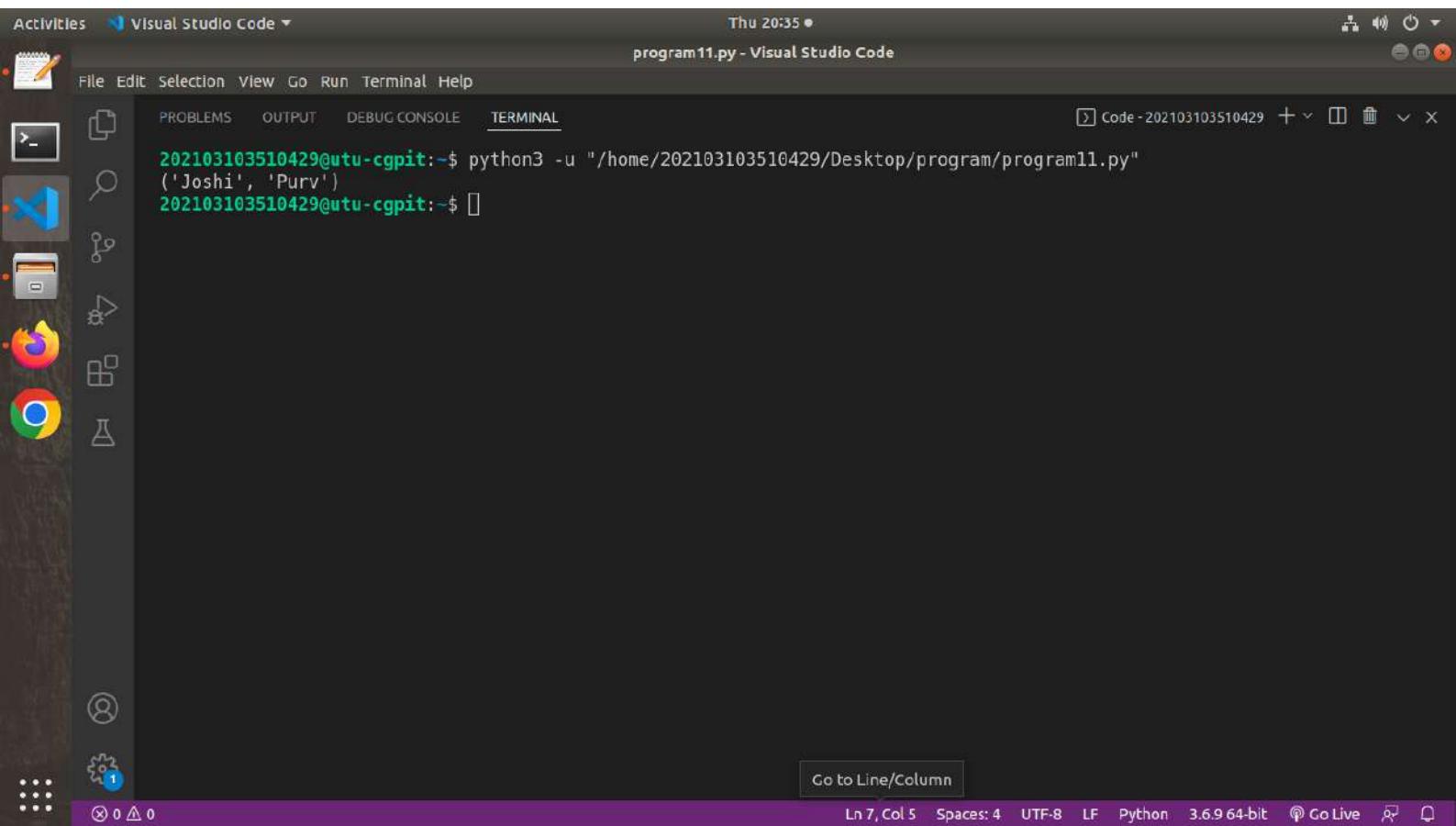
```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program10.py"
1.Addtion
2.Multiple
3.Divide
4.Odd or Even
5.Subtraction
choice your number= 1
enter the value=2
3
5
8
202103103510429@utu-cgpit:~$
```

Ln 45, Col 1 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 11

Write a python program to demonstrate the use of arbitrary arguments.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 def name(*list):
6     return list
7
8
9 list=("Joshi","Purv")
10 print(name(*list))
```



Practical 12

Create a class named student having attributes - std_name, std_age, std_branch, and std_city. Create a method named get_data() in student class that takes user input for these attributes and a method named display() that prints the attribute values on the terminal. Call both the methods by creating an instance std_obj of the class student.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenhai Joshi
4 class student:
5     std_name = None
6     std_age = None
7     std_branch = None
8     std_city = None
9     def get_data(self, name, age, branch, city):
10         self.std_name = name
11         self.std_age = age
12         self.std_branch = branch
13         self.std_city = city
14     def display(self):
15         print("Name:", self.std_name, "\nAge:", self.std_age, "\nBranch:", self.std_branch,
16         "\nCity:", self.std_city)
17 std_obj = student()
18 std_obj.get_data("Joshi Purv", "18", "B.tech CSE(SE)", "Rumla")
19 std_obj.display()
```

Activities Visual Studio Code

Thu 20:36 •

program12.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program12.py"
Name: Joshi Purv
Age: 18
Branch: B.tech CSE(SE)
City: Rumla
202103103510429@utu-cgpit:~$
```

Ln 18, Col 62 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 13

Write a python program to demonstrate basic banking operations. Create a class named banking having separate class methods for each operation. Call each method with an instance of the class and attribute values to be taken from the user.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 class BankAccount:
7     def __init__(self,acc_num,owner_name,balance) :
8
9
10         self.acc_num= acc_num
11         self.owner_name= owner_name
12         self.balance=balance
13
14     def deposite(self,dep_amount):
15         self.balance= self.balance + dep_amount
16
17         print("Your deposite amount is:",dep_amount)
18         print("Your current balance is :",self.balance)
19         print("\n")
20     def withdrawal(self,with_amount):
21         if(with_amount>self.balance):
22             print("Sorry!! check your bank balance are low")
23
24     else:
25
26         self.balance= self.balance - with_amount
27         print("Thank you !!! Your amount is credited:",with_amount)
28         print("Your current balance is:",self.balance)
29         print("\n")
30
31     def display(self):
32         print("Your account number is:",self.acc_num)
33         print("Your Bank Account name is:",self.owner_name)
34         print("Your current balance is :",self.balance)
35
36         print("\n")
37
38 acc_obj=BankAccount(12345677,"Purv Joshi",100000)
39 acc_obj.deposite(500)
40 acc_obj.withdrawal(2002)
41 acc_obj.display()
```

Activities Visual Studio Code Thu 20:38 •

program13.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program13.py"
Your deposite amount is: 500
Your current balance is : 100500

Thank you !!! Your amount is credited: 2002
Your current balance is: 98498

Your account number is: 12345677
Your Bank Account name is: Purv Joshi
Your current balance is : 98498

202103103510429@utu-cgpit:~$
```

Ln 10, Col 30 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 14

Create a class named employee having attributes - emp_name, emp_age, and emp_city. Create a method named get_data() in employee class that takes user input for these attributes. Derive a class named emp_derived() from the employee class, having an __init__() method that displays the attributes of the employee class upon instantiation.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 class employee:
6     emp_name = None
7     emp_age = None
8     emp_city = None
9     def get_data(self, name, age , city):
10         self.emp_name = name
11         self.emp_age = age
12         self.emp_city = city
13
14 class emp_derived(employee):
15     def __init__(self, obj):
16         print("Employee's Name:", obj.emp_name)
17         print("Employee's Age:", obj.emp_age)
18         print("Employee's City:", obj.emp_city)
19
20 emp1 = employee()
21 emp1.get_data(input("Enter Employee Name: "), input("Enter Employee's Age: "), input("Enter Employee's City: "))
22 emp_derived(emp1)
```

Activities Visual Studio Code Thu 20:39 •

program14.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program14.py"
Enter Employee Name: purv joshi
Enter Employee's Age: 18
Enter Employee's City: rumla
Employee's Name: purv joshi
Employee's Age: 18
Employee's City: rumla
202103103510429@utu-cgpit:~$
```

Ln 3, Col 22 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

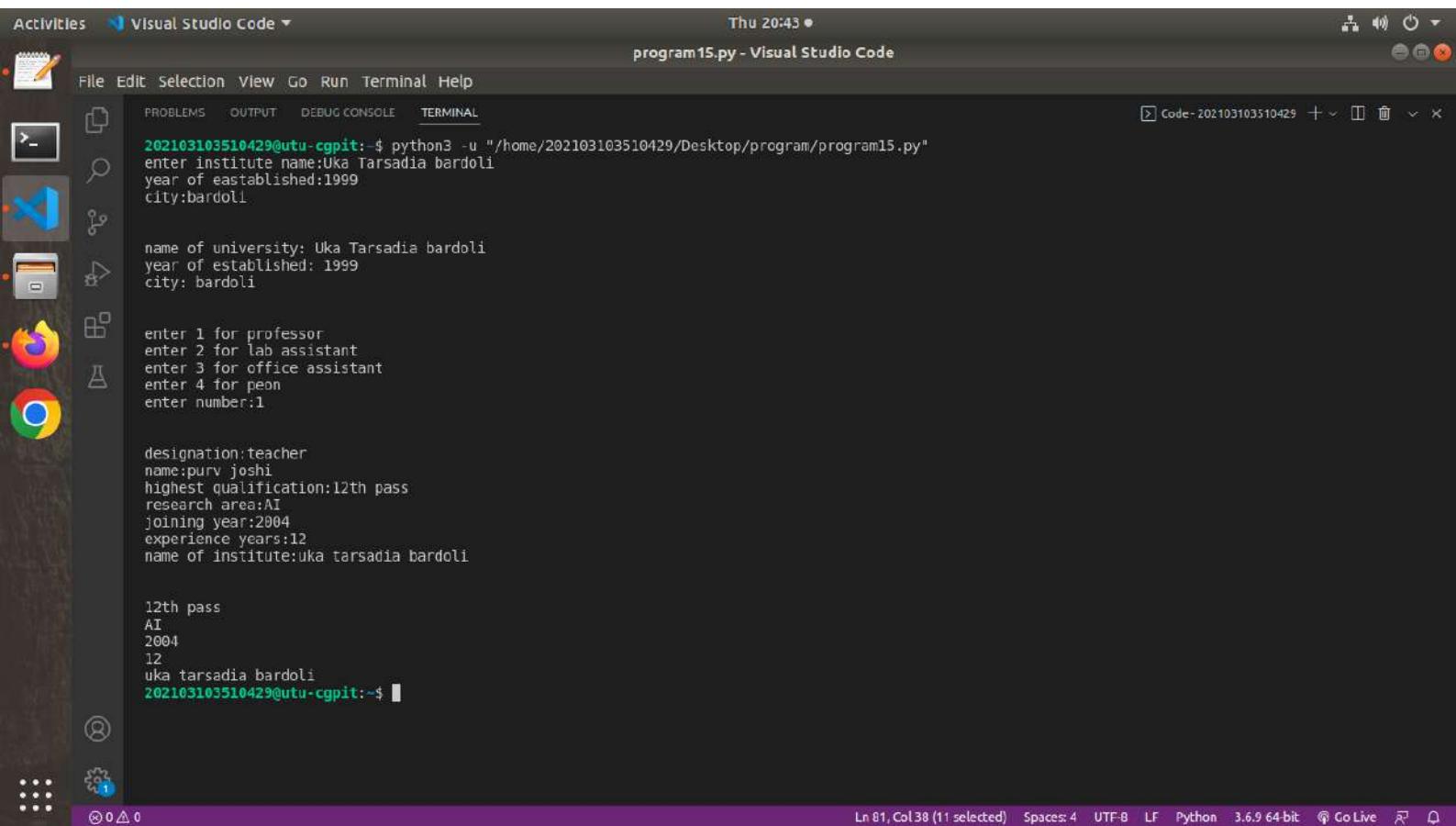
Practical 15

Create a base class named university with its attributes - name, year_of_estd, and city. Derive following class from the super class university: professor, lab_assistant, office_assistant, and peon. Make the program choice based for user. The attributes and method of various class are as below:

- Attributes of professor class: designation, highest_qualification, area_of_research, year_of_joining, year_of_experience, and name_of_institute.
- Methods of professor class: `__init__()` method that gets invoked upon instantiation and takes values of class attributes. The `display()` method that prints class attribute values along with attributes of its super class.
- Attributes of lab_assistant class: designation = “Lab Assistant” (static), highest_qualification, additiobnal_skilss, year_of_joining, and name_of_institue.
- Methods of lab_assistant class: `__init__()` method that gets invoked upon instantiation and takes values of class attributes. The `display()` method that prints class attribute values along with attributes of its super class.
- Attributes of office_assistant class: designation = “Office Assistant” (static), highest_qualification, year_of_joining, and name_of_institute. - Methods of office_assistant class: `__init__()` method that gets invoked upon instantiation and takes values of class attributes. The `display()` method that prints class attribute values along with attributes of its super class.
- Attributes of peon class: job_role = “office Peon” (static), qualification, year_of_joining, and name_of_institute.
- Methods of peon class: `__init__()` method that gets invoked upon instantiation and takes values of class attributes. The `display()` method that prints class attribute values along with attributes of its super class.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5 class university:
6     def __init__(self,name,year_of_estd,city):
7         self.name=name
8         self.year_of_estd=year_of_estd
9         self.city=city
10    def display(self):
11        print("\n")
12        print("name of university:",self.name)
13        print("year of established:",self.year_of_estd)
14        print("city:",self.city)
15 class professor(university):
16     def
17     __init__(self,designation,name,highest_qualification,area_of_research,year_of_joining,year_of_exp
18             self.designation=designation
19             self.name=name
20             self.highest_qualification=highest_qualification
21             self.area_of_research=area_of_research
22             self.year_of_joining=year_of_joining
23             self.year_of_experience=year_of_experience
24             self.name_of_institute=name_of_institute
25     def display(self):
26         print("\n")
27         print(self.highest_qualification)
28         print(self.area_of_research)
29         print(self.year_of_joining)
30         print(self.year_of_experience)
31         print(self.name_of_institute)
32 class lab_assistant(university):
33     designation="labour"
34     def __init__(self,highest_qualification,additional_skills,year_of_joining,name_of_institute):
35         self.highest_qualification=highest_qualification
36         self.additional_skills=additional_skills
37         self.year_of_joining=year_of_joining
38         self.name_of_institute=name_of_institute
39     def display(self):
40         print("\n")
41         print(self.highest_qualification)
42         print(self.additional_skills)
43         print(self.year_of_joining)
44         print(self.name_of_institute)
45
46 class office_assistant(university):
47     def __init__(self,highest_qualification,year_of_joining,name_of_institute):
48         self.highest_qualification=highest_qualification
49         self.year_of_joining=year_of_joining
50         self.name_of_institute=name_of_institute
51     def display(self):
52         print("\n")
53         print(self.highest_qualification)
54         print(self.year_of_joining)
55         print(self.name_of_institute)
56
```

```
57 class peon(university):
58     def __init__(self,qualification,year_of_joining,name_of_institute):
59         self.qualification=qualification
60         self.year_of_joining=year_of_joining
61         self.name_of_institute=name_of_institute
62     def display(self):
63         print("\n")
64         print(self.qualification)
65         print(self.year_of_joining)
66         print(self.name_of_institute)
67
68
69 obj1=university(input("enter institute name:"),int(input("year of eastablished:")),input("city:"))
70 obj1.display()
71 print("\n")
72 print("enter 1 for professor")
73 print("enter 2 for lab assistant")
74 print("enter 3 for office assistant")
75 print("enter 4 for peon")
76 class_input=int(input("enter number:"))
77 print("\n")
78
79
80 if class_input==1:
81     obj2=professor(input("designation:"),input("name:"),input("highest qualification:"),input("re-
area:"),int(input("joining year:")),int(input("experience years:")),input("name of institute:"))
82     obj2.display()
83 elif class_input==2:
84     obj3=lab_assistant(input("highest qualification:"),input("additional skills:"),int(input("yea-
joining:")),input("name of institute:"))
85     obj3.display()
86 elif class_input==3:
87     obj4=office_assistant(input("highest qualification:"),int(input("year of joining:")),input("n-
obj4.display()
88 elif class_input==4:
89     obj5=peon(input("highest qualification:"),int(input("year of joining:")),input("name of insti-
obj5.display()
```



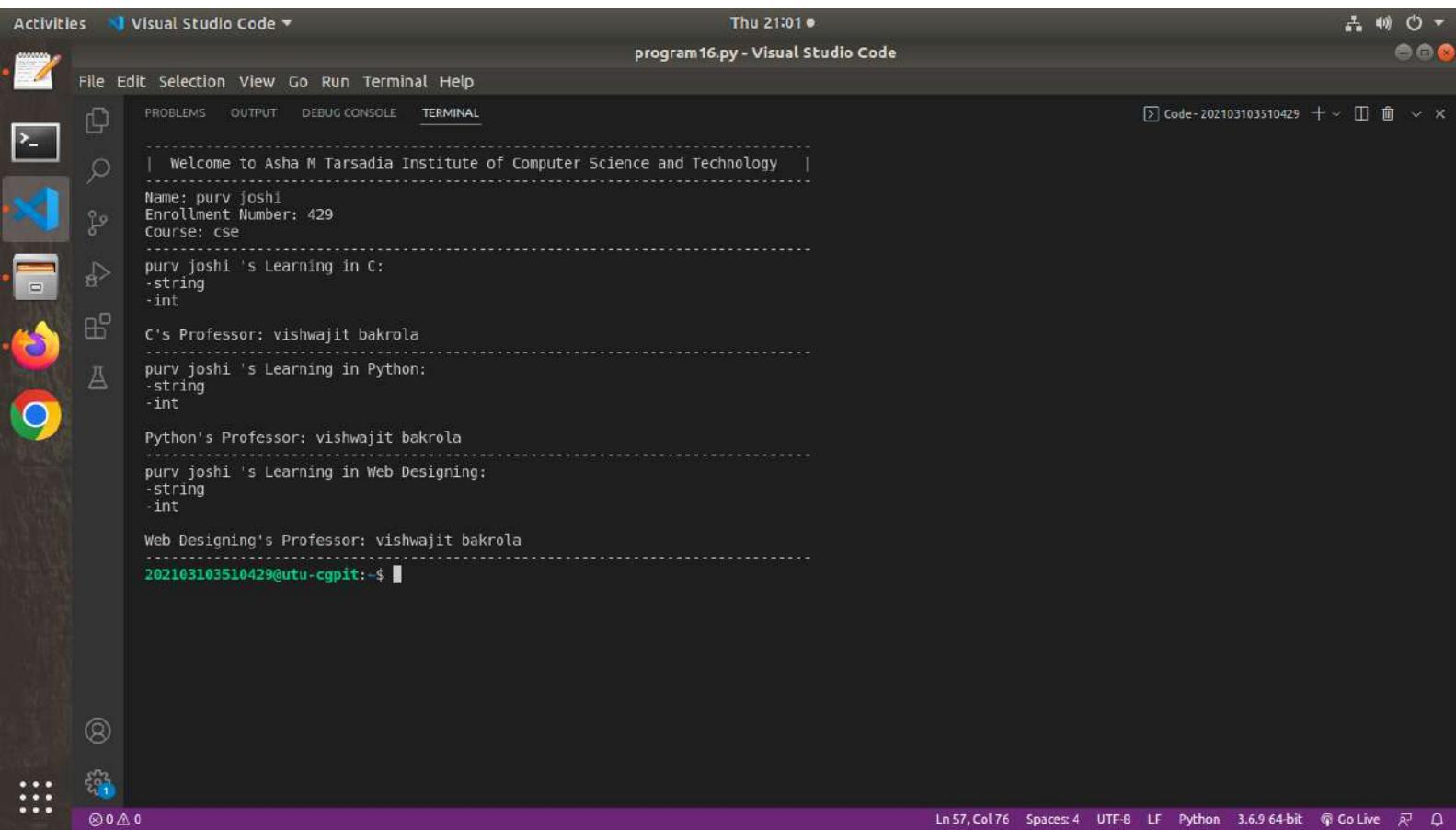
Practical 16

Create three classes named - *C*, *Python*, and *Web_Designing* each having two primary attributes as *learnings_* and *name_of_professor*. Derive a class named *student* from these classes. The *student* class has following methods and attributes:

- 1. Global *std_college* attribute with static values.**
- 2. *__init__()* method with attributes - *std_name*, *std_enrollment_no*, and *std_course*.**
- 3. *display()* method to display various attribute values of the terminal.**

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 class c:
7     def __init__(self, c_learning, c_professor):
8         self.c_learning = c_learning.replace(" ", "").split(",")
9         self.c_professor = c_professor
10 class python:
11     def __init__(self, py_learning, py_professor):
12         self.py_learning = py_learning.replace(" ", "").split(",")
13         self.py_professor = py_professor
14 class web_designing:
15     def __init__(self, wd_learning, wd_professor):
16         self.wd_learning = wd_learning.replace(" ", "").split(",")
17         self.wd_professor = wd_professor
18
19 class student(c, python, web_designing):
20     std_college = "Asha M Tarsadia Institute of Computer Science and Technology"
21     def __init__(self, std_name, std_enroll, std_course, c_learning, c_professor, py_learning,
22                  py_professor, wd_learning, wd_professor):
23         self.std_name = std_name
24         self.std_enroll = std_enroll
25         self.std_course = std_course
26         c.__init__(self, c_learning, c_professor)
27         python.__init__(self, py_learning, py_professor)
28         web_designing.__init__(self, wd_learning, wd_professor)
29     def display(self):
30         print("-----",
31             "\n| Welcome to", self.std_college, " |", "\n-----",
32             "\nName:", self.std_name, "\nEnrollment Number:",
33             self.std_enroll, "\nCourse:", self.std_course, "\n-----")
34         print(self.std_name, "'s Learning in C: ")
35         for i in self.c_learning:
36             print("-", end="")
37             print(i)
38         print("\nC's Professor:", self.c_professor, "\n-----")
39         print(self.std_name, "'s Learning in Python: ")
40         for i in self.py_learning:
41             print("-", end="")
42             print(i)
43         print("\nPython's Professor:", self.py_professor, "\n-----")
44         print(self.std_name, "'s Learning in Web Designing: ")
45         for i in self.wd_learning:
46             print("-", end="")
47             print(i)
48         print("\nWeb Designing's Professor:", self.wd_professor, "\n-----")
49
50 def main():
51     print("\033c", end="")
52     name = input("Enter student's name: ")
53     enroll = input("Enter enrollment number: ")
54     course = input("Enter course: ")
```

```
51 c_l = input("Enter C's learning of student (Seperated by comma): ")
52 c_p = input("Enter C professor's name: ")
53 py_l = input("Enter Python's learning of student (Seperated by comma): ")
54 py_p = input("Enter Python professor's name: ")
55 wd_l = input("Enter Web Designing's learning of student (Seperated by comma): ")
56 wd_p = input("Enter Web Designing professor's name: ")
57 std1 = student(name, enroll, course, c_l , c_p, py_l, py_p, wd_l, wd_p)
58 print("\033c", end="")
59 std1.display()
60 main()
```



Practical 17

Write a python program to demonstrate the use of data hiding.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenhai Joshi
4
5 class student:
6     __student_cepa=9.2
7     def display(self):
8         print(self.__student_cepa)
9
10 s=student()
11 s.display()
12 print(s.__student_cepa)
```

Activities Visual Studio Code ▾

Thu 21:06 •

program17.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program17.py"
9.2
Traceback (most recent call last):
  File "/home/202103103510429/Desktop/program/program17.py", line 12, in <module>
    print(s.__student_cepa)
AttributeError: 'student' object has no attribute '__student_cepa'
202103103510429@utu-cgpit:~$
```

Ln 14, Col 1 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 18

Write a python program to create a class named *area*. Define a class method *find_area()* that can find areas of different shapes whose value is given by the user. Invoke the class method by instantiation and prove method overloading.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 class area:
7     def __init__(self, len=None, wid=None):
8         self.len=len
9         self.wid=wid
10    def find_area(self):
11        if self.len!=None and self.wid==None :
12            print("Area of Square:",self.len**2)
13        elif self.len!=None and self.wid!=None :
14            print("Area of Rectangle: ",self.len*self.wid)
15
16 choice=input("Enter Square to get area of square OR Enter Rectangle to get the area of
17 Rectangle:")
18 if choice=="Square":
19     len=int(input("Enter the Side of Square: "))
20     obj1=area(len)
21     obj1.find_area()
22 elif choice=="Rectangle":
23     len=int(input("Enter the length of Rectangle: "))
24     wid=int(input("Enter the width of Rectangle: "))
25     obj1=area(len,wid)
26     obj1.find_area()
```

practical18.py - purv_program_py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Python + - □ ▾ ▾

PURV_PROGRAM_PY

- L.txt
- M.txt
- N.txt
- O.txt
- P.txt
- practical18.py**
- program2.png
- program23.png
- program23.py
- program23.txt
- program24.png
- program24.py
- program25.png
- program25.py
- program26.py
- program27.png
- program27.py
- Q.txt
- R.txt
- S.txt
- T.txt
- U.txt
- V.txt
- W.txt
- X.txt
- Y.txt
- Z.txt

> OUTLINE > TIMELINE

```
202103103510429@utu-cgpit:~/purv_program_py$ /bin/python3 /home/purvjoshi/purv_program_py/practical18.py
Enter Square to get area of square OR Enter Rectangle to get the area of Rectangle:Square
Enter the Side of Square: 2
Area of Square: 4
202103103510429@utu-cgpit:~/purv_program_py$ /bin/python3 /home/purvjoshi/purv_program_py/practical18.py
Enter Square to get area of square OR Enter Rectangle to get the area of Rectangle:Rectangle
Enter the length of Rectangle: 3
Enter the width of Rectangle: 5
Area of Rectangle: 15
202103103510429@utu-cgpit:~/purv_program_py$
```

Ln 3, Col 22 Spaces: 4 UTF-8 CRLF Python 3.10.4 64-bit ⌂ ⌂

Practical 19

Write a python program to demonstrate the use of method overriding.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 class base:
7     def display(self):
8         print("i am base class")
9
10 class derive(base):
11     def display(self):
12         print("i am derive class")
13         super().display()
14
15 ob=derive()
16 ob.display()
```

Activities Visual Studio Code

Thu 21:06 •

program19.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program19.py"
i am derive class
i am base class
202103103510429@utu-cgpit:~$ []
```

Ln 4, Col 1 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 20

Write a python program to demonstrate the use of try-catch block for exception handling.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 while(True):
7     try:
8         number=input("This program will throw exception and quit while entering anything other
than integer: ")
9         int(number)
10    except ValueError:
11        print("Enter a valid number, Press ctrl+c to quit the loop")
12        break
```

Activities Visual Studio Code Thu 21:08 •

program20.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program20.py"
This program will throw exception and quit while entering anything other than integer: 1
This program will throw exception and quit while entering anything other than integer: 2
This program will throw exception and quit while entering anything other than integer: 3
This program will throw exception and quit while entering anything other than integer: 23
This program will throw exception and quit while entering anything other than integer: hjg
Enter a valid number, Press ctrl+c to quit the loop
202103103510429@utu-cgpit:~$
```

Ln 4, Col 1 Tab Size: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 21

Write a python program to raise an exception with the python raise keyword.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 inp = int(input("Enter any number between 1 to 10: "))
7 if(inp<1 or inp>10):
8     raise Exception("You are supposed to enter a number between 1 to 10")
9 else:
10    for i in range(inp):
11        print(i+1)
```

Activities Visual Studio Code Thu 21:08 •

program21.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program21.py"
Enter any number between 1 to 10: 3
1
2
3
202103103510429@utu-cgpit:~$
```

Ln 4, Col 1 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 22

Write a python program to demonstrate the try-finally block.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 try:
7     int(input("Enter a Integer: "))
8 except ValueError:
9     print("Enter a valid Integer")
10 finally:
11     print("Program ended !")
```

Activities Visual Studio Code

Thu 21:09 •

program22.py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Code - 202103103510429 + ×

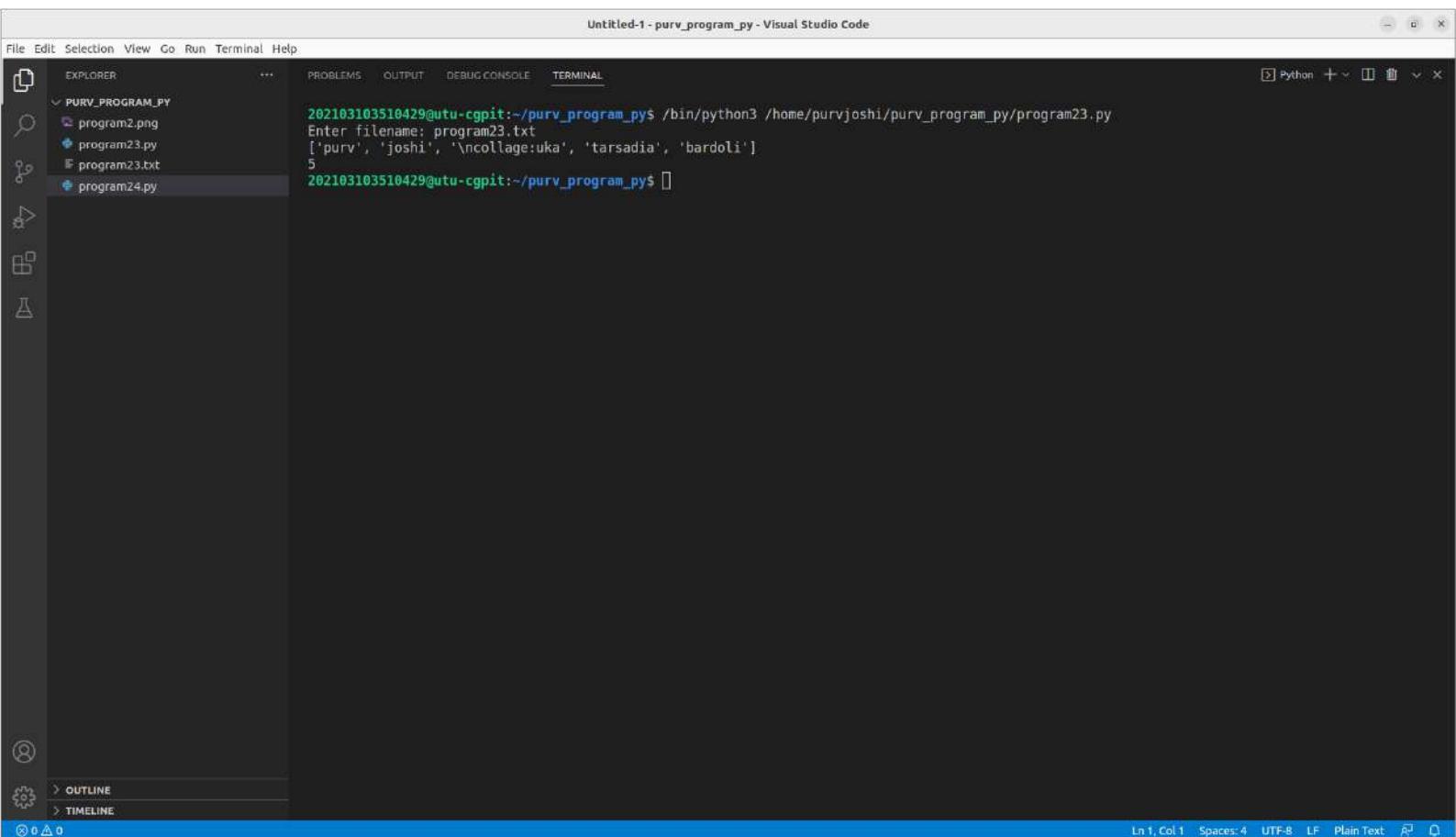
```
202103103510429@utu-cgpit:~$ python3 -u "/home/202103103510429/Desktop/program/program22.py"
Enter a Integer: 12
Program ended !
202103103510429@utu-cgpit:~$
```

Ln 7, Col 36 Spaces: 4 UTF-8 LF Python 3.6.9 64-bit Go Live

Practical 23

Write a python program to read the content of a file and return the number of words in a file to the user.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenhai Joshi
4 def cha_1(r):
5     for i in r:
6         n=r.count(i)
7         print(f"{i}={n} ")
8
9
10 file_name=input("Enter the file name = ")
11
12 comb=open(file_name,"r")
13 a=comb.read()
14 r=a.split()
15 print(cha_1(r))
```



Practical 24

Write a python program to read and show first N lines to the user. The number of lines N will be taken from user input.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenhai Joshi
4
5
6 file = input("Enter Filename: ")
7 file = open(file, "r")
8 lines = file.readlines()
9
10 opr = input("Enter Number of lines to be printed: ")
11
12 for i in range(int(opr)+1):
13     print(lines[i])
```

program24.py - purv_program_py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Python + ×

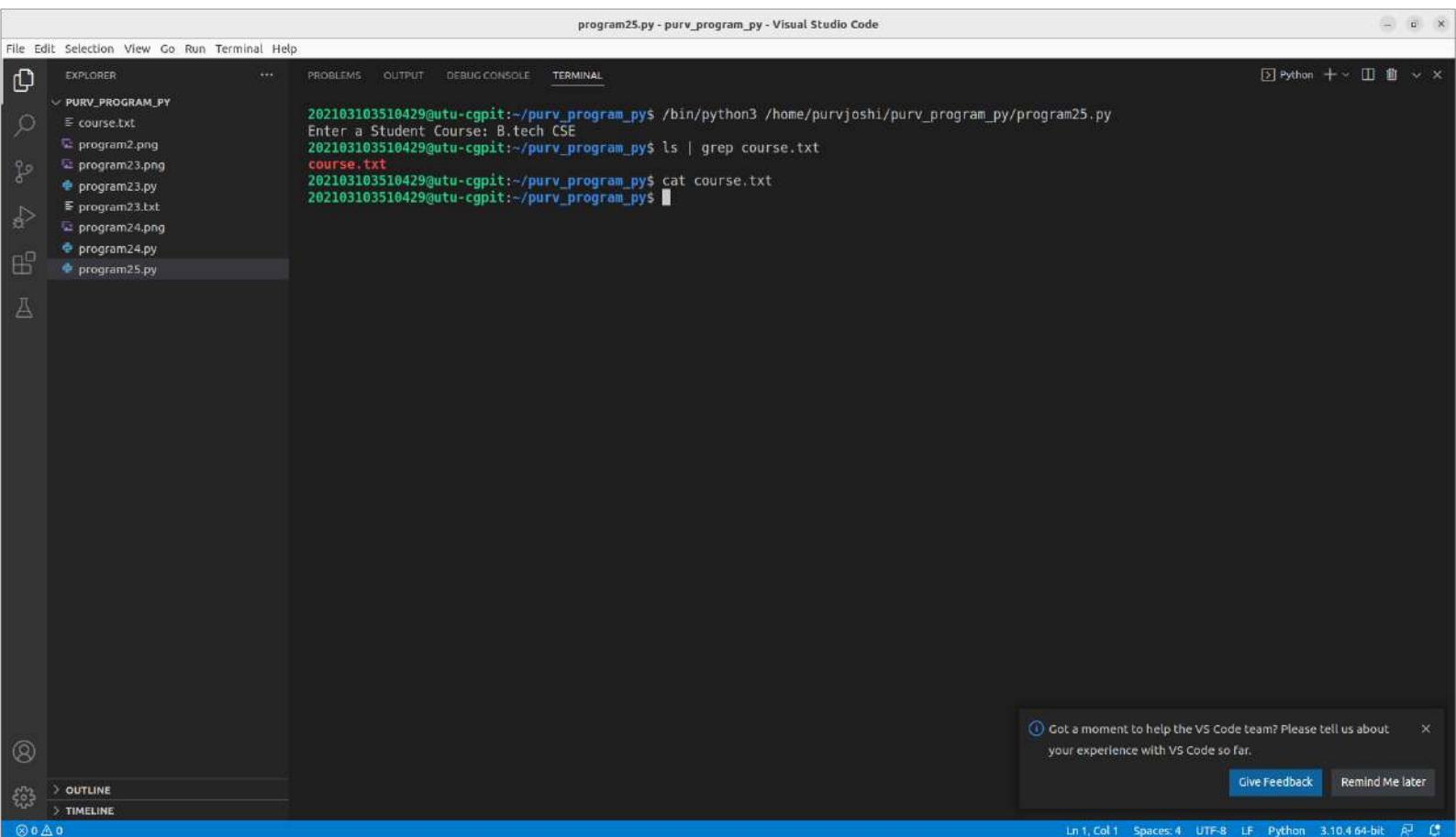
202103103510429@utu-cgpit:~/purv_program_py\$ /bin/python3 /home/purvjoshi/purv_program_py/program24.py
Enter Filename: program23.txt
Enter Number of lines to be printed: 1
purv joshi
collage:uka tarsadia bardoli
202103103510429@utu-cgpit:~/purv_program_py\$

Ln 13, Col 20 Tab Size: 4 UTF-8 LF Python 3.10.4 64-bit ⌂

Practical 25

**Write a python program that takes input of a student course from the user
and and write it in a file.**

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 course = input("Enter a Student Course: ")
7 fp = open("course.txt", "w")
8 fp.write(course)
```



Practical 26

Write a python program to copy the content of one file to another file.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 src = input("Name of source file: ")
7 des = input("Name of destination file: ")
8
9 src_op = open(src, "r")
10 des_op = open(des, "r")
11
12 des_op.write(src.read())
13
14 src_op.close()
15 des_op.close()
16 print("Operation Successful !")
```

program26.py - purv_program_py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Python + -

PURV_PROGRAM_PY

- 1.txt
- 2.txt
- A.txt
- B.txt
- C.txt
- course.txt
- D.txt
- E.txt
- F.txt
- G.txt
- H.txt
- I.txt
- J.txt
- K.txt
- L.txt
- M.txt
- N.txt
- O.txt
- P.txt
- practical18.py
- program2.png
- program18.png
- program23.png
- program23.py
- program23.txt
- program24.png
- program24.py
- program25.png

> OUTLINE > TIMELINE

```
202103103510429@utu-cgpit:~/purv_program_py$ /bin/python3 /home/purvjoshi/purv_program_py/program26.py
Name of source file: 1.txt
Name of destination file: 2.txt
Operation Successful !
202103103510429@utu-cgpit:~/purv_program_py$
```

Ln 10, Col 22 Spaces: 4 UTF-8 LF Python 3.10.4 64-bit ⚙

File Edit Selection View Go Run Terminal Help

EXPLORER ... program24.py program25.py program26.py 1.txt X pr ... 2.txt X

PURV_PROGRAM_PY

- 1.txt
- 2.txt
- A.txt
- B.txt
- C.txt
- course.txt
- D.txt
- E.txt
- F.txt
- G.txt
- H.txt
- I.txt
- J.txt
- K.txt
- L.txt
- M.txt
- N.txt
- O.txt
- P.txt
- practical18.py
- program2.png
- program18.png

OUTLINE

TIMELINE 1.txt

No timeline information was provided.

① 0 ▲ 0

Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text ⌂ ⌂

The screenshot shows the Visual Studio Code interface with two open text files. The left sidebar displays a file tree under 'PURV_PROGRAM_PY' with numerous text files (A.txt through P.txt) and other files like 'practical18.py', 'program2.png', and 'program18.png'. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The status bar at the bottom shows 'Ln 1, Col 1' and file settings for 'Spaces: 4', 'UTF-8', 'LF', and 'Plain Text'. The main area contains two tabs: '1.txt - purv_program_py - Visual Studio Code' and '2.txt'. Both tabs show the same content:

```
1 hello purv joshi!
```

Practical 27

Write a python program that creates 26 text files named A.txt, B.txt, and up to Z.txt.

```
1 #B.Tech Computer Science And Technology
2 #202103103510429
3 #Purv Devenbhai Joshi
4
5
6 import string
7
8 for i in range(len(string.ascii_uppercase)):
9     open(string.ascii_uppercase[i] + ".txt", "w")
```

program27.py - purv_program_py - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Python + × ☰

PURV_PROGRAM_PY

```
202103103510429@utu-cgpit:~/purv_program_pys /bin/python3 /home/purvjoshi/purv_program_py/program27.py
202103103510429@utu-cgpit:~/purv_program_pys program27.py
program27.py: command not found
202103103510429@utu-cgpit:~/purv_program_pys ls
A.txt      C.txt  F.txt  I.txt  L.txt  O.txt      program23.txt  program25.png  program27.py  Q.txt  T.txt  W.txt  Z.txt
B.txt      D.txt  G.txt  J.txt  M.txt  program23.png  program24.png  program25.py  program27.png  R.txt  U.txt  X.txt
course.txt E.txt  H.txt  K.txt  N.txt  program23.py  program24.py  program26.py  P.txt      S.txt  V.txt  Y.txt
202103103510429@utu-cgpit:~/purv_program_pys$
```

Q.txt
R.txt
S.txt
T.txt
U.txt
V.txt
W.txt

OUTLINE TIMELINE

Ln 11, Col 5 Spaces: 4 UTF-8 LF Python 3.10.4 64-bit ⚙