



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

S.E/SEM IV/CBCGS/AIML

Academic Year: 2021-22

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLLNO.	AIML51
SUBJECT	PYTHON LAB
COURSECODE	CSL405
PRACTICALNO.	01
DOP	31/01/2022
DOS	21/03/2022

Name :- Singh Sudham Dharmendra.

Branch :- CSE (AI & ML)

Roll no :- AIML - 51

Subject :- Python Programming.

Topic :- Experiment No :- 1

Date of submission :- 21st March 2022.

Aim:- Program Using basic data types (Numeric (int, float), list, tuple, set, dictionaries and string).

- Take two numbers as input (float) & print addition.
- Swap two numbers.
- Solve the quadratic equation $ax^2 + bx + c = 0$.
- Perform (at least 5) operations on each
 - list
 - tuple
 - set
 - dictionaries
 - strings

Theory :-

① List :- A list is created by placing elements in square brackets `[]` separated by commas.

Eg:- `my_list [1, 2, 'sudham']`

② Tuple :- Tuples are used to store multiple items in a single value
Eg:- `tuple = (5, 9, 11, 3)`

③ Set :- A set is an unordered collection of items. A set is created by placing all elements inside curly braces `{ }` . Separated by comma or by using the built-in set () function
Eg:- `My_set = {1, 2, 3}`

④ Dictionaries :- Creating a dictionary is as simple as placing items inside curly braces `{ }` separated by commas. An item has a key and a corresponding value that is expressed as `key: value`
eg:- `my_diet = {1: 'apple', 2: 'orange'}`

⑤ String:- String can be created by enclosing a characters inside a Single quotes or double quotes.
Eg:- `my_string = "Hello world"`

main.py

Run Shell Clear

```
1 #Take two numbers as input(float) and add them.
2 num1 = input('Enter first number: ')
3 num2 = input('Enter second number: ')
4 sum = float(num1) + float(num2)
5 print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
6
7
8
9
10
```

Enter first number: 55.5
Enter second number: 85
The sum of 55.5 and 85 is 140.5
>

main.py

Run Shell Clear

```
1 #To swap two numbers
2 x = 55
3 y = 125
4 temp = x
5 x = y
6 y = temp
7 print('The value of x after swapping: {}'.format(x))
8 print('The value of y after swapping: {}'.format(y))
9
10
11
12
13
14
```

The value of x after swapping: 125
The value of y after swapping: 55
>|

main.py

Run Shell Clear

```
1 #Solve the quadratic equation
2 #i
3 import cmath
4 a = float(input('Enter a: '))
5 b = float(input('Enter b: '))
6 c = float(input('Enter c: '))
7 d = (b**2) - (4*a*c) # calculate the discriminant
8 sol1 = (-b-cmath.sqrt(d))/(2*a)
9 sol2 = (-b+cmath.sqrt(d))/(2*a)
10 print('The solution are {0} and {1}'.format(sol1,sol2))
11
12
13
14
1c
```

Enter a: 2
Enter b: 3
Enter c: 4
The solution are (-0.75-1.1989578808281798j) and (-0.75+1.1989578808281798j)
>

main.py

```
1 #Perform atleast 5 operations on each
2 #list
3 l1 = [1,2,3,4,5]
4 l2 = [51, 'sudham', 937]
5 print(l1)
6 print(len(l1))
7 l2.append(78)
8 print(l2)
9 print(l1 + l2)
10 print(l1.pop())
11 del l1[1]
12 print(l1)
13
14
15
16
17
```

Run Shell

```
[1, 2, 3, 4, 5]
5
[51, 'sudham', 937, 78]
[1, 2, 3, 4, 5, 51, 'sudham', 937, 78]
5
[1, 3, 4]
>|
```

Clear

main.py

```
1 #tuple
2 tuple1 = (1,2,3)
3 tuple2 = ('LTCE', 'Koparkharine')
4 print(tuple1)
5 i = tuple1.index(2)
6 print(tuple1)
7 print(type(tuple2))
8 n_tuple = ("cat", [8, 4, 6], (1, 2, 3))
9 n_tuple = ("cat", [8, 4, 6], (1, 2, 3))
10 print(tuple1[-1])
11
12
13
14
15
```

Run Shell

```
(1, 2, 3)
(1, 2, 3)
<class 'tuple'>
3
>
```

Clear

main.py

```
1 #set
2 my_set = {1, 3, 5, "sudham", 6}
3 print(my_set)
4 my_set.add(2)
5 print(my_set)
6 my_set.update([4, 5], {1, 6, 8})
7 print(my_set)
8 my_set.discard(2)
9 print(my_set.pop())
10 print(my_set)
11 my_set.clear()
12 print(my_set)
13
14
15
```

Run Shell

```
{1, 3, 5, 6, 'sudham'}
{1, 2, 3, 5, 6, 'sudham'}
{1, 2, 3, 4, 5, 6, 'sudham', 8}
1
{3, 4, 5, 6, 'sudham', 8}
set()
>|
```

Clear

main.py

Run Shell Clear

```
1 #dictionaries
2 my_dict = {'Name': 'sudahm singh', 'Age': 20, 1:4, 3:6, 5:8}
3 print(my_dict['Name'])
4 print(my_dict.get('Age'))
5 my_dict['Address'] = 'Kalyan(East)'
6 print(my_dict)
7 print(my_dict.pop(1))
8 print(my_dict)
9 del my_dict
10
11
12
13
14
15
```

sudahm singh
20
{'Name': 'sudahm singh', 'Age': 20, 1: 4, 3: 6, 5: 8, 'Address': 'Kalyan(East)'}
4
{'Name': 'sudahm singh', 'Age': 20, 3: 6, 5: 8, 'Address': 'Kalyan(East)'}

main.py

Run Shell Clear

```
1 #strings
2 str = 'sudham singh'
3 print('str = ', str)
4 print('str[0] = ', str[0])
5 print('str[-1] = ', str[-1])
6 print('str[1:5] = ', str[1:5])
7 print('str[5:-2] = ', str[5:-2])
8 print('str * 3 =', str * 3)
9 list_enumerate = list(enumerate(str))
10 print('list(enumerate(str)) = ', list_enumerate)
11 print('len(str) = ', len(str))
```

str = sudham singh
str[0] = s
str[-1] = h
str[1:5] = udha
str[5:-2] = m sin
str * 3 = sudham singhsudham singhsudham singh
list(enumerate(str)) = [(0, 's'), (1, 'u'), (2, 'd'), (3, 'h'), (4, 'a'), (5, 'm'), (6, ' '), (7, 's'), (8, 'i'), (9, 'n'), (10, 'g'), (11, 'h')]
len(str) = 12



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

S.E/SEM IV/CBCGS/AIML

Academic Year: 2021-22

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLLNO.	AIML51
SUBJECT	PYTHON LAB
COURSECODE	CSL405
PRACTICALNO.	02
DOP	08/02/2022
DOS	21/03/2022

Name :- Singh Sudham Dharmendra.

Branch :- CSE (AI & ML)

Roll no:- AIML 51

Subject :- Python Programming.

Topic:- Experiment No:- 2

Date of Submission :- 21st March 2022.

Aim :- Program using different control statements.

- (a) smallest of 3 numbers (using nested if else)
- (b) To check if the input number is prime or not (using For loop)
- (c) To check if number provided by the user is Armstrong number or not (using while loop)

Theory :-

- ① While loop :- Repeat a statement or group of statement while a given condition is True. It tests the condition before executing the loop body.
- ② For loop :- Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
- ③ nested loops :- You can use one or more loop inside another while, for or do while loop.
- ④ break statement :- Terminates the loop statement and transfers execution to the statement immediately following the loop.
- ⑤ Continue statement :- Causes the loop to skip the remainder of its body and immediately retests its condition prior to reiterating.
- ⑥ Pass statement :- It is used when a statement is required syntactically but you do not want any command or code to execute.

main.py

```
1 #Smallest of 3 numbers(using nested if else)
2 def smallest(x, y, z):
3     if x <= y:
4         if x <= z:
5             min = x
6         else:
7             min = z
8     else:
9         if y <= z:
10            min = y
11     else:
12        min = z
13 print("Smallest number among", x,".",y,"and",z,"is: ",min)
14
15 smallest(200, 321, 250)
16 smallest(30, 20, 70)
```

Shell

```
Smallest number among 200 , 321 and 250 is: 200
Smallest number among 30 , 20 and 70 is: 20
>
```

main.py

```
1 #To check if the input number is prime or not using for loop
2 num = int(input("Enter a number: "))
3 if num > 1:
4     for i in range(2,num):
5         if (num % i) == 0:
6             print(num,"is not a prime number")
7             print(i,"times",num//i,"is",num)
8             break
9     else:
10    print(num,"is a prime number")
11 else:
12    print(num,"is not a prime number")
13
14
15
16
```

Shell

```
Enter a number: 2
2 is a prime number
>
```

main.py

```
1 #To check if number provided by the user is armstrong number or not using while
loop
2 num = int(input("Enter a number: "))
3 sum = 0
4 temp = num
5 while temp > 0:
6     digit = temp % 10
7     sum += digit ** 3
8     temp //= 10
9 if num == sum:
10    print(num,"is an Armstrong number")
11 else:
12    print(num,"is not an Armstrong number")
```

Shell

```
Enter a number: 407
407 is an Armstrong number
>
```



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

S.E/SEM IV/CBCGS/AIML

Academic Year: 2021-22

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLLNO.	AIML51
SUBJECT	PYTHON LAB
COURSECODE	CSL405
PRACTICALNO.	03
DOP	14/02/2022
DOS	21/03/2022

Name :- Singh Sudham Dharmandra.

Branch :- CSE (AI & ML)

Roll no:- AIML - 51

Subject:- Python Programming.

Topic :- Experiment No. 3

Date of submission :- 21st March 2022.



- Aim:- Program using concept of functions, classes & objects
- i - a) Make a simple calculations that can add, subtract, multiply & divide using function.
 - b) To find largest number from the list.
 - (i) Using normal function
 - (ii) Using Lambda function.
 - ii - a) Write a program to print Employee information (use object variables & class variables).

Theory:-

① Functions:- A function is a two block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

Syntax: `def function_name (parameters);
 "function - docstring"
 function - suite
 return [expression]`

② Class:- A class is a user defined blue print as prototype from which objects are created. Each class instance can have attributes attached to it for maintaining its state. Class instance can also have methods for modifying their state.

Syntax:- `class className
 # Statement-1
 :
 :
 # Statement N`



③ Object :- An object is an instance of a class. A class is like a blueprint while an instance is a copy of the class with actual values. An object consists of -

- (i) state - It is represented by the attributes of an object.
It also reflects the properties of an object.
- (ii) Behaviours - It is represented by the methods of an object.
It also reflects the response of an object to other objects.
- (iii) Identity :- It gives a unique name to an object and enables one object to interact with other objects.

```
exp3.py > ...
1  #Make a simply calculator that can add, subtract,multiply and divide using functions.
2  def add(x, y):
3      return x + y
4  def subtract(x, y):
5      return x - y
6  def multiply(x, y):
7      return x * y
8  def divide(x, y):
9      return x / y
10 print("Select operation.")
11 print("1.Add")
12 print("2.Subtract")
13 print("3.Multiply")
14 print("4.Divide")
15 while True:
16     choice = input("Enter choice(1/2/3/4): ")
17     if choice in ('1', '2', '3', '4'):
18         num1 = float(input("Enter first number: "))
19         num2 = float(input("Enter second number: "))
20         if choice == '1':
21             print(num1, "+", num2, "=", add(num1, num2))
22         elif choice == '2':
23             print(num1, "-", num2, "=", subtract(num1, num2))
24         elif choice == '3':
25             print(num1, "*", num2, "=", multiply(num1, num2))
26         elif choice == '4':
27             print(num1, "/", num2, "=", divide(num1, num2))
28         next_calculation = input("Let's do next calculation? (yes/no): ")
29         if next_calculation == "no":
30             break
31     else:
32         print("Invalid Input")
```

Ln 32, Col 31 Spaces: 4 UTF-8 CRLF Python 3.7.6 64-bit ⌂ ⌂ Prettier ⌂ ⌂

PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS D:\vs\code\mydirectory> python -u "d:\vs\code\mydirectory\exp3.py"
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 1
Enter first number: 45
Enter second number: 5
45.0 + 5.0 = 50.0
Let's do next calculation? (yes/no): yes
Enter choice(1/2/3/4): 4
Enter first number: 2
Enter second number: 1
2.0 / 1.0 = 2.0
Let's do next calculation? (yes/no): no
PS D:\vs\code\mydirectory>
```

```
exp3.py > ...
1 #To find the largest number from the list using normal function
2 def myMax(list1):
3     max = list1[0]
4     for x in list1:
5         if x > max :
6             max = x
7     return max
8 list1 = [23, 45, 12, 95, 34]
9 print("Largest element is:", myMax(list1))
```

PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE

PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\exp3.py"
Largest element is: 95
PS D:\vs code\mydirectory>

Ln 9, Col 43 Spaces: 4 UTF-8 CRLF Python 3.7.6 64-bit ⚡ Prettier ⚡

```
exp3.py > ...
1 #To find the largest number from the list using the lamda function
2 list = [222,333,444,555,666,999]
3 print(max(list, key=lambda value: int(value)))
```

PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE

PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\exp3.py"
999
PS D:\vs code\mydirectory>

Ln 3, Col 48 Spaces: 4 UTF-8 CRLF Python 3.7.6 64-bit ⚡ Prettier ⚡

A screenshot of a VS Code terminal window. The terminal tab is selected at the top. The code editor shows a Python script named `exp3.py` with the following content:

```
1 #Write a program to print employee information use object variables and class variables.
2 class Employee:
3     Name = "SUDHAM SINGH"
4     Age = 20
5     Id = 2412
6 e = Employee()
7 print(e.Name)
8 print(e.Age)
9 print(e.Id)
10 print(type(e))
```

The terminal output shows the execution of the script:

```
PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\exp3.py"
SUDHAM SINGH
20
2412
<class '__main__.Employee'>
PS D:\vs code\mydirectory>
```

At the bottom of the terminal window, there are status indicators: Ln 8, Col 13, Spaces: 4, UTF-8, CRLF, Python 3.7.6 64-bit, Prettier, and a refresh icon.



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

S.E/SEM IV/CBCGS/AIML

Academic Year: 2021-22

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLL NO.	AIML51
SUBJECT	PYTHON LAB
COURSE CODE	CSL405
PRACTICAL NO.	04
DOP	21/02/2022
DOS	24/03/2022

Name:- Singh Sudham Dharmendra.

Branch:- CSE (AI & ML)

Roll no:- AIML - 51

Subject:- Python Programming

Topic:- Experiment No. 4

Date of Submission :- 24/03/22



Aim:- Program for file handling & directories

- Statement:-
- a) Program that count the number of characters in a text file.
 - b) To count number of words in a text file.
 - c) A program to count the number of lines in a text file.
 - d) To display file available in current directory.

Theory:-

① File handling :- Python too supports file handling and allows users to handle file i.e to ~~to read & write files~~ along with many other file handling options to operate on files. There are three kinds of mode, that Python provides & how files can be opened.

"r" for reading

"w" for writing.

"a" for appending

"x" for both reading & writing.

So the syntax being : Open (filename, mode)

② Directories:- All files are contained with various directories and python has no problem handling these too. The os module has several methods that help you create, remove and change directories.

The mkdir() method:- You can use the mkdir() method of the os module to create directories in the current directory.

Syntax: os.mkdir ("newdir")

The chdir() method:- You can use the chdir() method to change the current directory.

Syntax : os.chdir ("new dir")

The getcwd() method:- The getcwd() method displays the current working directory.

Syntax : os.getcwd()

The rmdir() method:- The rmdir() method deletes the directory which is passed as an argument in the method. Before removing a directory, all the contents in it should be removed.

Syntax : os.rmdir (dirname)

The screenshot shows a dark-themed VS Code interface. In the top-left, there's a file icon followed by 'expt4.py > ...'. Below it is the Python code:

```
1 #Program that count the number of characters in a text file.
2 file = open("C:\\Users\\duste\\Downloads\\sudham\\data.txt", "r")
3 data = file.read()
4 number_of_characters = len(data)
5 print(f"Number of characters in text file {number_of_characters}")
```

At the bottom of the editor, the status bar shows 'PROBLEMS 4' and other tabs like 'OUTPUT', 'TERMINAL', and 'DEBUG CONSOLE'. The terminal tab is active, displaying the command 'PS D:\\vs code\\mydirectory> python -u "d:\\vs code\\mydirectory\\expt4.py"' and its output 'Number of characters in text file 20'. The status bar at the bottom also includes icons for Code, Output, Terminal, and Debug.

This screenshot shows the same VS Code environment as the first one, but with different Python code in the editor:

```
1 #To count number of words in a text file.
2 file = open("c:\\Users\\duste\\Downloads\\sudham\\data.txt", "rt")
3 data = file.read()
4 words = data.split()
5 print(f"Number of words in text file : {len(words)}")
```

The terminal tab shows the command 'PS D:\\vs code\\mydirectory> python -u "d:\\vs code\\mydirectory\\expt4.py"' and its output 'Number of words in text file : 4'. The status bar at the bottom includes icons for Code, Output, Terminal, and Debug, along with specific file statistics: 'Ln 5, Col 55', 'Spaces: 4', 'UTF-8', 'CRLF', '(Python)', 'Prettier', and 'Format'.

```
 expt4.py > ...
1 #A program to count the number of lines in a text file
2 fname = input("Enter file name: ")
3 num_lines = 0
4 with open(fname, 'r') as f:
5     for line in f:
6         num_lines += 1
7 print("Number of lines:")
8 print(num_lines)

PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE
PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\expt4.py"
Enter file name: C:\\Users\\dust\\Downloads\\sudham\\data.txt
Number of lines:
1
PS D:\vs code\mydirectory>
```

```
 expt4.py > ...
1 #To display file available in current directory
2 import glob
3 print("List of All Files in Current Directory:")
4 for file in glob.glob("*.*"):
5     print(file)

PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE
PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\expt4.py"
List of All Files in Current Directory:
arrayminmax.class
arrayminmax.java
ascii.class
ascii.java
asciinput.class
asciinput.java
asciloop.class
asciloop.java
B.class
BINARy.c
BINARy.exe
currency.java
data.class
data.java
dlca2.c
dlca2.exe
expt4.py
expt6.c
expt6.exe
first.py
infinix-postfix.c
infinix-postfix.exe
infix.c
infix.exe
lengthbreadth.class
LinkedList.c
LinkedList.exe
monday.c
monday.exe
MyClass.class
MyClass.java
newoddeven.class

Ln 4, Col 30  Spaces: 4  UTF-8  CRLF  {} Python  ⚡  ⚡ Prettier  ⚡  ⚡
```



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

**S.E/SEM IV/CBCGS/AIML
Academic Year: 2021-22**

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLL NO.	AIML51
SUBJECT	PYTHON LAB
COURSE CODE	CSL405
PRACTICAL NO.	05
DOP	14/03/2022
DOS	24/03/2022

Name:- Singh Sudham Dharmendra

Branch:- CSE (AI & ML)

Roll no:- AIML - 51

Subject:- Python Programming

Topic:- Experiment No. 5

Date of Submission:- 24/03/2022

Aim:- Program to demonstrate i) Stack ii) Queue (iii) linked list.

Theory:-

① Stack: A stack is a linear data structure that stores items in a LIFO & FIFO manner. In stack a new element is added at one end and an element is removed from that end only. The insert and delete operations are after called push & pop.

② Queue: Like stack, queue is a linear data structure that stores items in FIFO manner with a queue the least recently added item is removed first. A good example of queue is any queue of consumers from a resource where the consumer that come first is served first.

③ Linked list: Like arrays, linked list is a linear data structure. Unlike arrays linked list elements are not stored at a contiguous location; the elements are linked using pointers.

The screenshot shows a terminal window in VS Code with the following content:

```
expt5.py > ...
1 #Program to demonstrate stack
2 stack = []
3 a = int(input("Enter the size of stack : "))
4 for x in range(0,a):
5     b = int(input())
6     stack.append(b)
7 print(stack)
8 stack.append(8)
9 print(stack)
10 stack.pop()
11 print(stack)
12 print(len(stack))

PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE
```

TERMINAL

```
PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\expt5.py"
Enter the size of stack : 5
1
2
3
4
5
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5, 8]
[1, 2, 3, 4, 5]
5
PS D:\vs code\mydirectory>
```

Bottom status bar: Line 12, Col 18 Spaces: 4 UTF-8 CRLF {} Python ⌂ Prettier ⌂ ⌂

The screenshot shows a code editor window with a dark theme. On the left is a vertical file navigation bar with icons for file operations like New, Open, Save, and Close. The main area contains a Python script named `expt5.py`. The code defines a `Queue` class with methods for enqueueing and dequeuing values, and a loop for interacting with the user. The code editor has a status bar at the bottom with information about the file's location (D:\vs code\mydirectory), line count (Ln 21), column count (Col 25), and encoding (UTF-8). It also includes icons for Python, Prettier, and other tools.

```
❶ expt5.py > ...
1 #Program to demonstrate queue
2 class Queue:
3     def __init__(self):
4         self.items = []
5     def is_empty(self):
6         return self.items == []
7     def enqueue(self, data):
8         self.items.append(data)
9     def dequeue(self):
10        return self.items.pop(0)
11 q = Queue()
12 while True:
13     print('enqueue <value>')
14     print('dequeue')
15     print('quit')
16     do = input('What would you like to do? ').split()
17     operation = do[0].strip().lower()
18     if operation == 'enqueue':
19         q.enqueue(int(do[1]))
20     elif operation == 'dequeue':
21         if q.is_empty():
22             print('Queue is empty.')
23         else:
24             print('Dequeued value: ', q.dequeue())
25     elif operation == 'quit':
26         break
```

The screenshot shows a terminal window with a dark theme. At the top, there are tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active. The terminal window displays the command `python -u "d:\vs code\mydirectory\expt5.py"` followed by the program's output. The output shows the user interacting with the queue, enqueueing values 23, 45, and 67, dequeuing value 23, and then exiting. The terminal has a status bar at the bottom with the same information as the code editor: file location, line count, column count, and encoding.

```
PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\expt5.py"
enqueue <value>
dequeue
quit
What would you like to do? enqueue 23 45 67
enqueue <value>
dequeue
quit
What would you like to do? dequeue 23
Dequeued value: 23
enqueue <value>
dequeue
quit
What would you like to do? dequeue 67
Queue is empty.
enqueue <value>
dequeue
quit
What would you like to do? quit
PS D:\vs code\mydirectory>
```

The screenshot shows a code editor window with a dark theme. The code is written in Python and defines a linked list structure. It includes classes for nodes and a linked list, methods for appending data and displaying the list, and a main block that creates a linked list, adds 5 elements, and prints the result.

```
1 #Program to demonstrate linked list
2 class Node:
3     def __init__(self, data):
4         self.data = data
5         self.next = None
6 class LinkedList:
7     def __init__(self):
8         self.head = None
9         self.last_node = None
10    def append(self, data):
11        if self.last_node is None:
12            self.head = Node(data)
13            self.last_node = self.head
14        else:
15            self.last_node.next = Node(data)
16            self.last_node = self.last_node.next
17    def display(self):
18        current = self.head
19        while current is not None:
20            print(current.data, end = ' ')
21            current = current.next
22 a_llist = LinkedList()
23 n = int(input('How many elements would you like to add? '))
24 for i in range(n):
25     data = int(input('Enter data item: '))
26     a_llist.append(data)
27 print("The linked list: ", end = '')
28 a_llist.display()
```

The screenshot shows a terminal window with a dark theme. It displays the command to run the Python script, the user's input for the number of elements, the individual data inputs, the resulting linked list, and the final command prompt.

```
PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\expt5.py"
How many elements would you like to add? 5
Enter data item: 23
Enter data item: 45
Enter data item: 12
Enter data item: 56
Enter data item: 87
The linked list: 23 45 12 56 87
PS D:\vs code\mydirectory>
```



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

**S.E/SEM IV/CBCGS/AIML
Academic Year: 2021-22**

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLL NO.	AIML51
SUBJECT	PYTHON LAB
COURSE CODE	CSL405
PRACTICAL NO.	06
DOP	21/03/2022
DOS	26/03/2022



Name :- Singh Sudham Dharmendra.

Branch : CSE (AI & ML)

Roll no:- AIML - S1

Subject :- Python Programming.

Topic :- Experiment No. 6

Date of Submission :- 26/03/2022.



Aim:- Program to demonstrate use of Numpy : Array objects.
Exploring basics of Numpy methods.

Theory :-

Numpy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform and matrices.

Numpy stands for Numerical Python.

Numpy is often used along with packages like Scipy (Scientific Python) and Matplotlib (plotting library).

Array objects :- Numpy provides an N-dimensional array type the ndarray, which describes a collection of "items" of the same type. All ndarrays are homogeneous, i.e; every item takes up the same size block of memory and all blocks are interpreted in exactly the same way.

Array Indexing :-

Slicing :- Just like list in python, Numpy arrays can be sliced. As arrays can be multidimensional, you need to specify a slice for each dimension of array.

OUTPUT:

```
❶ expt6.py > ...
1 import numpy as np
2 print ('Array of 1-8')
3 arr=np.array([1,2,3,4,5,6,7,8])
4 print(arr)
5 print(arr)
6 print('Element at index 0')
7 print (arr[0])
8 print ('Sum of elements at index 2 and 3')
9 print(arr[2]+[3])
10 print('Elements from index 1-5')
11 print (arr[1:5])
12 print('Elements from index 3 from the end to index 1 from the array')
13 print(arr[-3:-1])
14 print('Make a view, change the view, and display both arrays')
15 x=arr.view()
16 x[5]=31
17 print (arr)
18 print (x)
19 print("Print the value of the base attribute to check if an arrays data or not")
20 x= arr.copy()
21 y=arr.view()
22 print(x.base)
23 print(y.base)
24 print('New Array for reshape')
25 arr= np.arange(12)
26 print(arr)
27 print('Reshaping array to 4x3 matrix')
28 newarr=arr.reshape(4,3)
29 print(newarr)
```

Ln 29, Col 14 Spaces: 4 UTF-8 CRLF Python 3.7.6 64-bit ⌂ ⌂ Prettier ⌂ ⌂

```
PS D:\vs code\mydirectory> python -u "d:\vs code\mydirectory\expt6.py"
Array of 1-8
[1 2 3 4 5 6 7 8]
[1 2 3 4 5 6 7 8]
Element at index 0
1
Sum of elements at index 2 and 3
[6]
Elements from index 1-5
[2 3 4 5]
Elements from index 3 from the end to index 1 from the array
[6 7]
Make a view, change the view, and display both arrays
[ 1 2 3 4 5 31 7 8]
[ 1 2 3 4 5 31 7 8]
Print the value of the base attribute to check if an arrays data or not
None
[ 1 2 3 4 5 31 7 8]
New Array for reshape
[ 0 1 2 3 4 5 6 7 8 9 10 11]
Reshaping array to 4x3 matrix
[[ 0 1 2]
 [ 3 4 5]
 [ 6 7 8]
 [ 9 10 11]]
```

PS D:\vs code\mydirectory> █

Ln 29, Col 14 Spaces: 4 UTF-8 CRLF Python 3.7.6 64-bit ⌂ ⌂ Prettier ⌂ ⌂

```
Python 3.6.9 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: /home/computer/aiml51.py =====
=====
Array a: [[4 8]
 [9 9]]
Array b: [[1 6]
 [6 7]]
a>b
[[ True  True]
 [ False False]]
[[5 14]
 [15 16]]
[[3 2]
 [3 2]]
>>>
===== RESTART: /home/computer/aiml51.py =====
=====
Array a: [[4 8]
 [9 9]]
Array b: [[1 6]
 [6 7]]
a>b
[[ True  True]
 [ True  True]]
a<b
[[ False False]
 [ False False]]
[[5 14]
 [15 16]]
[[3 2]
 [3 2]]
[[ 4 48]
 [54 63]]
>>>

File Edit Format Run Options Window Help
print('elements from index 3 from the end to the index 1 from the ar
print(arr[-3:-1])
print('make a view, change the view, and display both teh array')
x = arr.view()
x[5]=31
print(x)
print(x)
print("print the value of the base attribute to check if an array")
x=marr.copy()
y=marr.view()
print(x.base)
print(y.base)
print('new array for reshape')
arr=np.arange(12)
print(arr)
print('reshaping the array to 4x3 matrix')
newarr=arr.reshape(4,3)
print(newarr)'''

import numpy as np
a= np.array([[4,8],[9,9]])
b= np.array([[1,6],[6,7]])
print ("Array a:", a)
print ("Array b:", b)

'''comparison = a==b
equal_arrays = comparison.all()
print(equal_arrays)'''
print("a>b")
print(np.greater(a,b))

print("a<b")
print(np.greater(b,a))

print(a+b)
print(a-b)
print(a*b)

Ln: 209 Col: 4
Ln: 47 Col: 9
```



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

S.E/SEM IV/CBCGS/AIML

Academic Year: 2021-22

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLL NO.	AIML51
SUBJECT	PYTHON LAB
COURSE CODE	CSL405
PRACTICAL NO.	07
DOP	21/03/2022
DOS	26/03/2022



Name:- Singh Sudham Dhamendra

Branch:- CSE(AI & ML)

Roll no:- AIML - 51

Subject:- Python Programming

Topic:- Experiment No. 7

Date of submission:- 26/03/2022

Aim:- Program to demonstrate Data series & Data frames using
Name:- Pandas.

Theory:- Pandas is an open-source library that is made mainly for working with relational or labelled data both easily and intuitively. It provides various data structures & operations for manipulating numerical data & time series. Pandas is fast and it has high performance & productivity for users.

Dataframe :-

Pandas DataFrame is a two dimensional size mtable, potentially heterogeneous tabular data structure i.e; data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consist of three principal components, the data, rows and columns.

Pandas DataFrame can be created from the lists, dictionary, and from a list of dictionaries etc.

Matplotlib:-

Matplotlib is a low level graph plotting library in python that serves a visualization utility. It is an open source & we can use it freely.

OUTPUT:

Activities Toplevel ▾

Mon 3:51 PM

File Edit Format Run Options Window Help

```
import pandas as pd

df= pd.DataFrame ({
    'ROLLNO' :[1,2,3,4,5],
    'NAME' :['A','B','C','D','E'],
    'PERCENT' :[95,96,97,98,97]
})
print(df)
```

Python 3.6.9 Shell

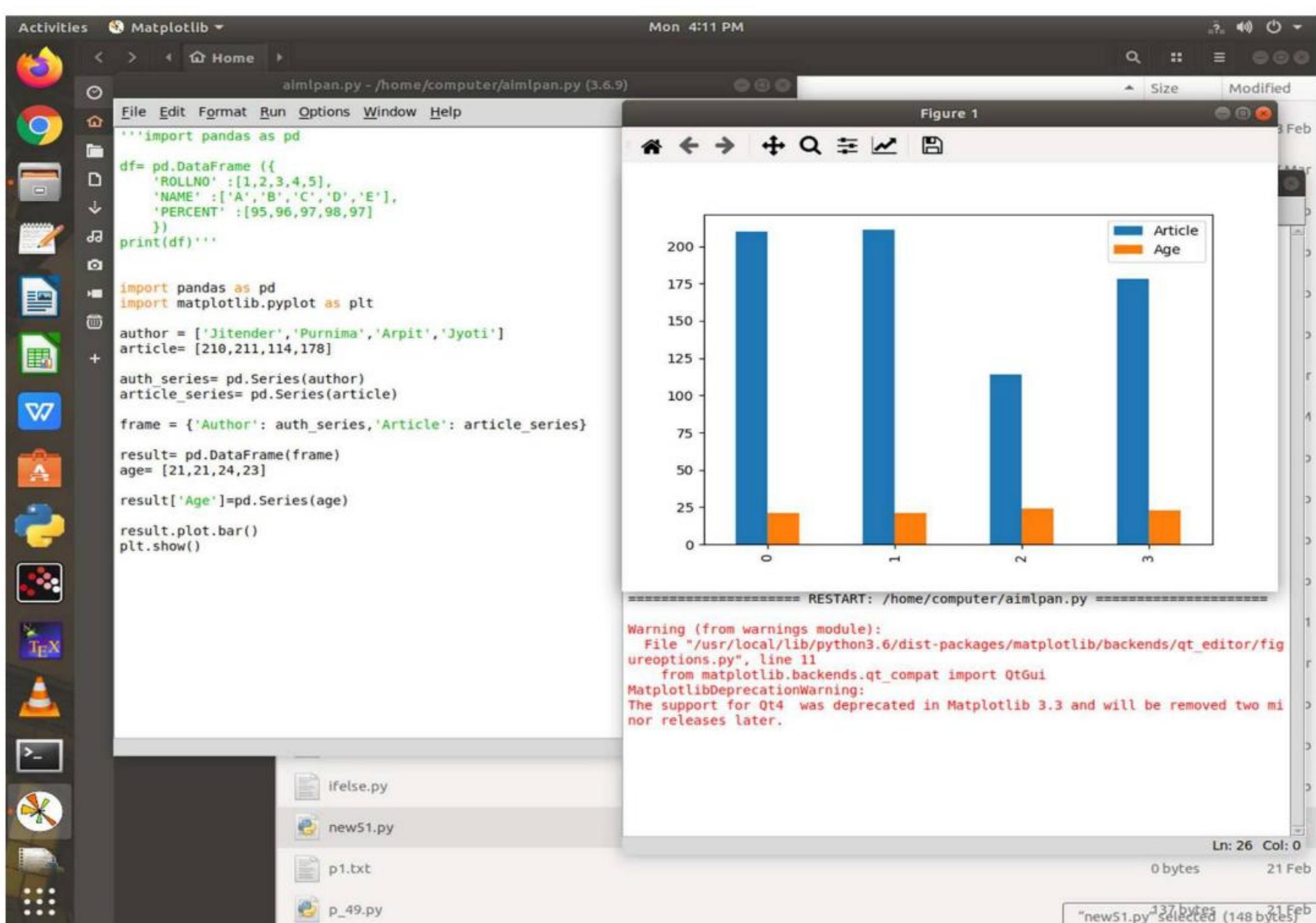
```
Python 3.6.9 (default, Dec  8 2021, 21:08:43)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: /home/computer/aimlpan.py =====
Traceback (most recent call last):
  File "/home/computer/aimlpan.py", line 1, in <module>
    import Pandas as pd
ModuleNotFoundError: No module named 'Pandas'
>>>
=====
RESTART: /home/computer/aimlpan.py =====
Traceback (most recent call last):
  File "/home/computer/aimlpan.py", line 5, in <module>
    'NAME' :[ A,B,C,D,E],
NameError: name 'A' is not defined
>>>
=====
RESTART: /home/computer/aimlpan.py =====
   NAME  PERCENT  ROLLNO
0      A       95        1
1      B       96        2
2      C       97        3
3      D       98        4
4      E       97        5
>>> |
```

ifelse.py
new51.py
p1.txt
p_49.py

Ln: 24 Col: 4

0 bytes 21 Feb

"new51.py" selected (148 bytes) 21 Feb





**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

S.E/SEM IV/CBCGS/AIML Academic Year: 2021-22

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLL NO.	AIML51
SUBJECT	PYTHON PROGRAMMING
COURSE CODE	CSL405
PRACTICAL NO.	08
DOP	28/03/2022
DOS	20/04/2022



Name :- Singh Sudham Dharmandra

Branch :- CSE (AI & ML)

Roll no:- AIML-51

Subject:- Python Programming

Topic:- Experiment No:- ⑧

Date of submission:- 20/04/2022

Aim:- Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes.

Theory:-

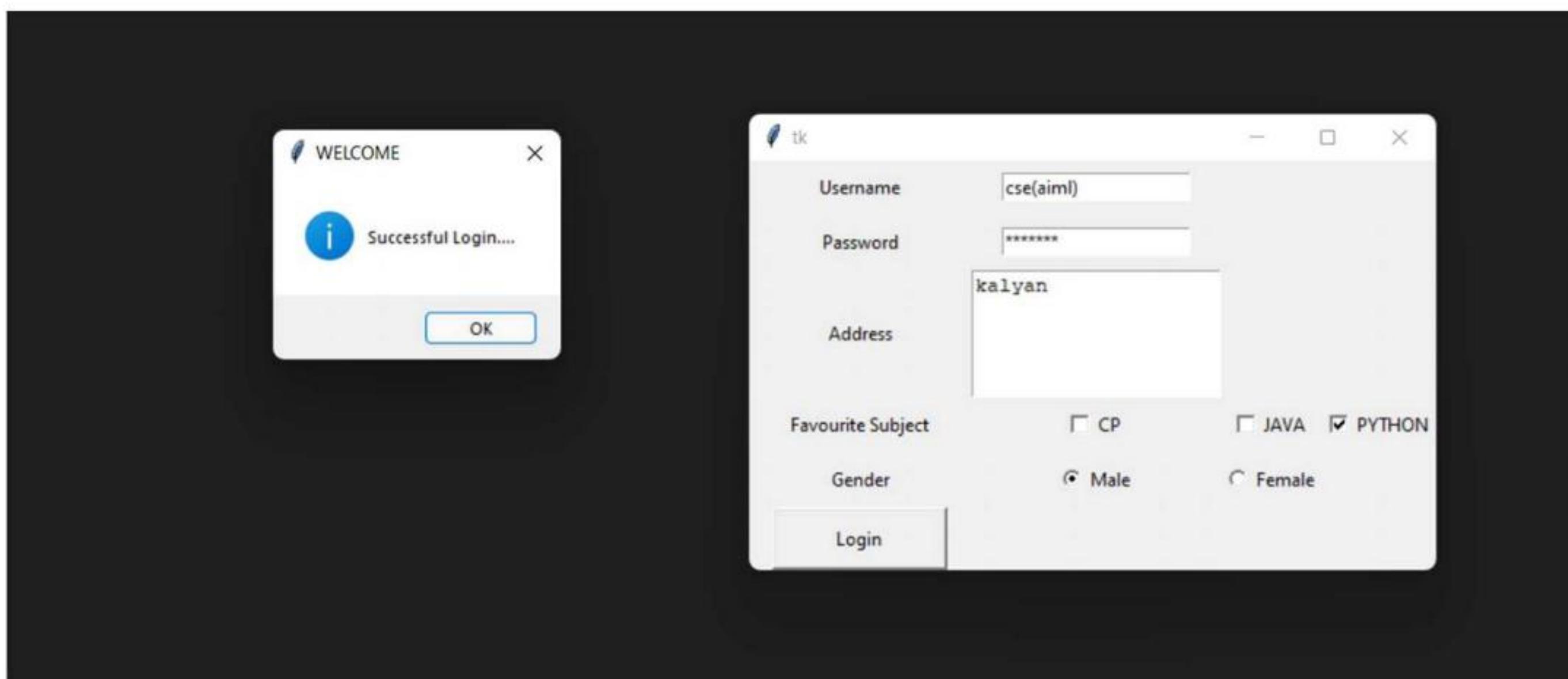
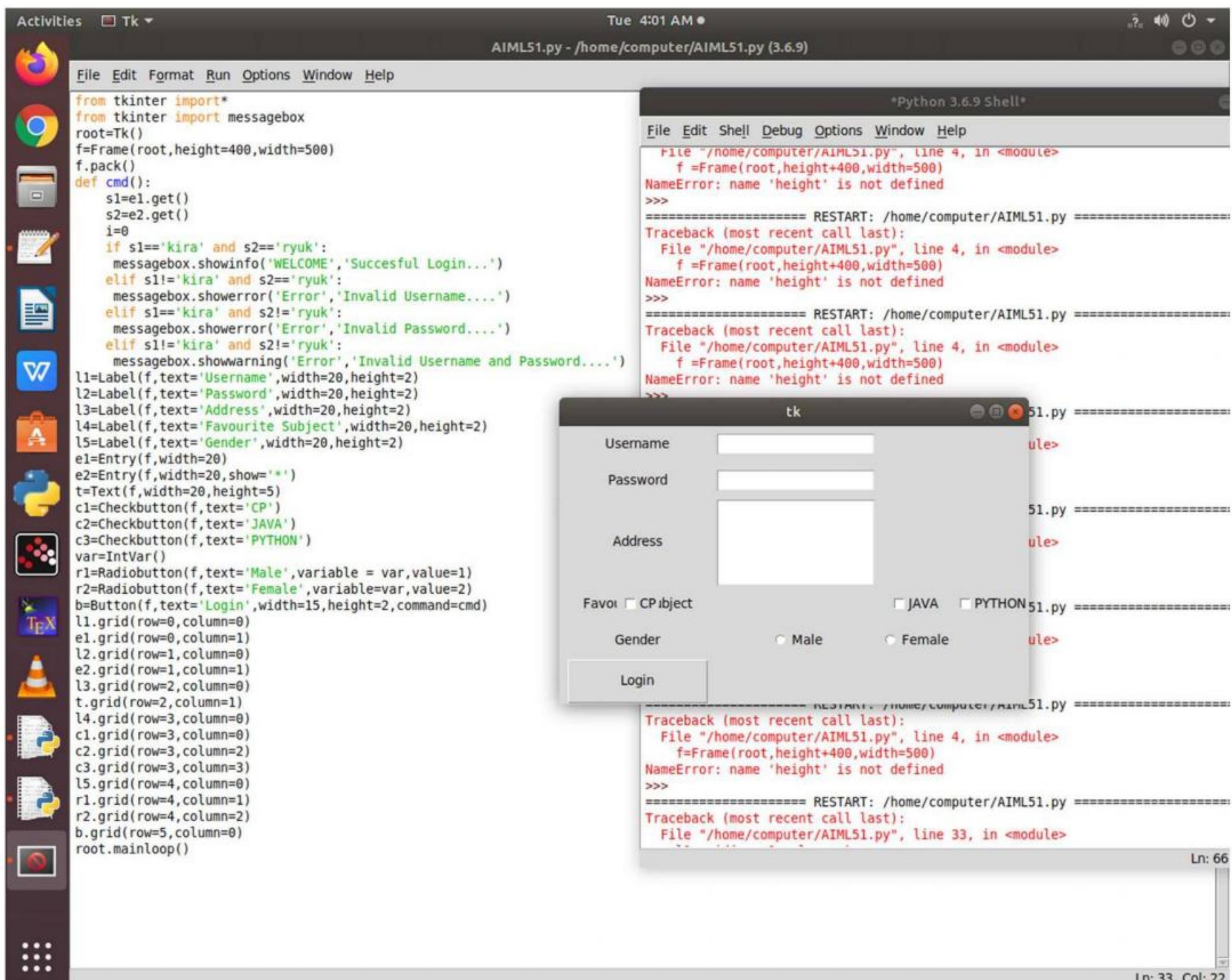
Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, Tkinter is the most commonly used method. It is a standard python interface to the Tk GUI toolkit shipped with python. Python with Tkinter is the fastest & easiest way to create the GUI applications.

Creating a GUI using Tkinter is an easy task.

There are a number of widgets which you can use in your Tkinter applications. Some of them are :-

- ① Button:- To add a button in any applications, this widget is used.
- ② Checkbutton:- To select any number of options by displaying a number of options to a user as toggle buttons.
- ③ Entry:- It is used to input the single line text entry from the user, for multi-line text input, text widget is used.
- ④ Frame:- It acts as a container to hold the widgets. It is used for grouping & organizing widgets.

⑤ Radiobutton:- It is used to offer multi-choice option to the user. It offers several options to the users & the user has to choose one option.





DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

S.E/SEM IV/CBCGS/AIML Academic Year: 2021-22

NAME	SINGH SUDHAM DHARMENDRA
BRANCH	CSE-(AI&ML)
ROLL NO.	AIML51
SUBJECT	PYTHON PROGRAMMING
COURSE CODE	CSL405
PRACTICAL NO.	09
DOP	20/04/2022
DOS	20/04/2022



Name :- Singh Sudham Dhawendra

Branch :- CSE (AI & ML)

Roll no. AIML-51

Subject:- Python Programming

Topic :- Experiment No:- 9

Date of submission. :- 20/4/2022



Aim:- Write a program to demonstrate CRUD (Create, Read, update and delete) operations on database [SQLite / MySQL].

Theory :-

MySQL -

MySQL is a relational database management system based on SQL - Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL, however, is for the purpose of a web database.

• Connecting MySQL Database in Python :-

We can connect MySQL database with a python program which enables us to manipulate the SQL database using python commands.

• Connect () -

The connect() constructor creates a connection to the MySQL server and returns a MySQL connection object.

• Conclusion -

We have successfully connected the SQL database with python and also manipulated the data stored in the SQL database.

CODE –

```

import pymysql as sql
mydb = sql.connect(host='localhost', user='root', password='', database='emp')
mycursor = mydb.cursor()
while True:
    print("\t menu \n1.create data\n2.read data\n3.update data\n4.delete data\n5.exit")
    ch = int(input("enter your choice"))
    if ch == 1:
        emp_id = input("enter the employee id : ")
        emp_name = input("enter the employee name : ")
        emp_age = int(input("enter the employee age : "))
        branch = input("enter the employee branch : ")
        salary = int(input("enter the employee salary : "))
        q = "insert into `empinfo` values ('"+emp_id+"','"+emp_name+"','"+str(emp_age)+"','"+branch+"','"+str(salary)+"');"
        mycursor.execute(q)
    elif ch == 2:
        print("the data in the table is")
        q = "select * from empinfo;"
        mycursor.execute(q)
        print(mycursor.fetchall())
        for i in mycursor:
            print(i + "\n")
    elif ch == 3:
        print("\tupdating the salary of a person - ")
        emp_id = input("enter the employee id : ")
        sal = int(input("enter the new salary of the person : "))
        q = "UPDATE `empinfo` SET salary =" + str(sal) + " WHERE emp_id = '" + emp_id + "';"
        mycursor.execute(q)
        mydb.commit()
        print(mycursor.rowcount, "updated successfully")
    elif ch == 4:
        print("\tdeleting a employee record - ")
        id = input("enter the employee id : ")
        q = "delete from empinfo where emp_id ='" + id + "';"
        mycursor.execute(q)
        mydb.commit()
    elif ch == 5:
        print("byee")
        break
    else:
        print("invalid choice !!!")
        continue

```

OUTPUT –