JEPPIAAR
ENGINEERING COLLEGE

DEPARTMENT

OF

COMPUTER SCIENCE & ENGINEERING

LAB MANUAL

GE3171

PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY

I YEAR - BATCH: 2021 -25

(All Departments)

Vision of Institution

To build Jeppiaar Engineering College as an Institution of Academic Excellence in Technical education and Management education and to become a World Class University.

Mission of Institution

M1	To excel in teaching and **learning, research and innovation** by promoting the principles of scientific analysis and creative thinking
M2	To participate in the production, **development and dissemination of knowledge** and interact with **national and international communities**
M3	To equip students with **values, ethics and life skills** needed to enrich their lives and enable them to meaningfully contribute to the **progress of society**
M4	To prepare students **for higher studies and lifelong learning**, enrich them with the **practical and entrepreneurial skills** necessary to excel as future professionals and contribute to **Nation's economy**

Program Outcomes (POs)

PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Vision of Department

To emerge as a globally prominent department, developing ethical computer professionals, innovators and entrepreneurs with academic excellence through quality education and research**.**

Mission of Department

M1	To create **computer professionals** with an ability to identify and **formulate the engineering problems** and also to provide **innovative solutions** through **effective teaching learning process.**
M2	To **strengthen the core-competence** in computer science and engineering and to create an ability to **interact** effectively with industries.
M3	To produce engineers with good professional skills, **ethical values** and life skills for the **betterment of the society.**
M4	To encourage students towards **continuous and higher level learning** on technological advancements and provide a platform for **employment and self-employment.**

Program Educational Objectives (PEOs)

PEO1	To address the real time complex engineering problems using innovative approach with strong core computing skills.
PEO2	To apply core-analytical knowledge and appropriate techniques and provide solutions to real time challenges of national and global society
PEO3	Apply ethical knowledge for professional excellence and leadership for the betterment of the society.
PEO4	Develop life-long learning skills needed for better employment and entrepreneurship

Program Specific Outcomes (PSOs)

Students will be able to

PSO1	An ability to understand the core concepts of computer science and engineering and to enrich problem solving skills to analyze, design and implement software and hardware based systems of varying complexity.
PSO2	To interpret real-time problems with analytical skills and to arrive at cost effective and optimal solution using advanced tools and techniques.
PSO3	An understanding of social awareness and professional ethics with practical proficiency in the broad area of programming concepts by lifelong learning to inculcate employment and entrepreneurship skills.

GE3171 PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY

COURSE OUTCOMES

CO1: Develop algorithmic solutions to simple computational problems
CO2: Develop and execute simple Python programs.
CO3: Implement programs in Python using conditionals and loops for solving problems.
CO4: Deploy functions to decompose a Python program.
CO5: Process compound data using Python data structures.
CO6: Utilize Python packages in developing software applications.

LIST OF EXERCISES

1. Identification and solving of simple real life or scientific or technical problems, and developing flow charts for the same. (Electricity Billing, Retail shop billing, Sin series, weight of a motorbike, Weight of a steel bar, compute Electrical Current in Three Phase AC Circuit, etc.)

2. Python programming using simple statements and expressions (exchange the values of two variables, circulate the values of n variables, distance between two points).

3. Scientific problems using Conditionals and Iterative loops. (Number series, Number Patterns, pyramid pattern)

4. Implementing real-time/technical applications using Lists, Tuples. (Items present in a library/Components of a car/ Materials required for construction of a building –operations of list & tuples)

5. Implementing real-time/technical applications using Sets, Dictionaries. (Language, components of an automobile, Elements of a civil structure, etc.- operations of Sets & Dictionaries

6. Implementing programs using Functions. (Factorial, largest number in a list, area of shape)

7. Implementing programs using Strings. (reverse, palindrome, character count, replacing characters)

8. Implementing programs using written modules and Python Standard Libraries (pandas, numpy. Matplotlib, scipy)

9. Implementing real-time/technical applications using File handling. (copy from one file to another, word count, longest word)

10. Implementing real-time/technical applications using Exception handling. (divide by zero error, voter's age validity, student mark range validation)

11. Exploring Pygame tool.

12. Developing a game activity using Pygame like bouncing ball, car race etc.

GE3171 PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY

INDEX

Ex.No: 1 **Constructing Flowcharts for Technical Problems**

Date :

1. **Problem statement :** Identification and solving of simple real life or scientific or technical problems, and developing flow charts for Electricity Billing, Retail shop billing, Sin series, weight of a motorbike, Weight of a steel bar, compute Electrical Current in Three Phase AC Circuit.

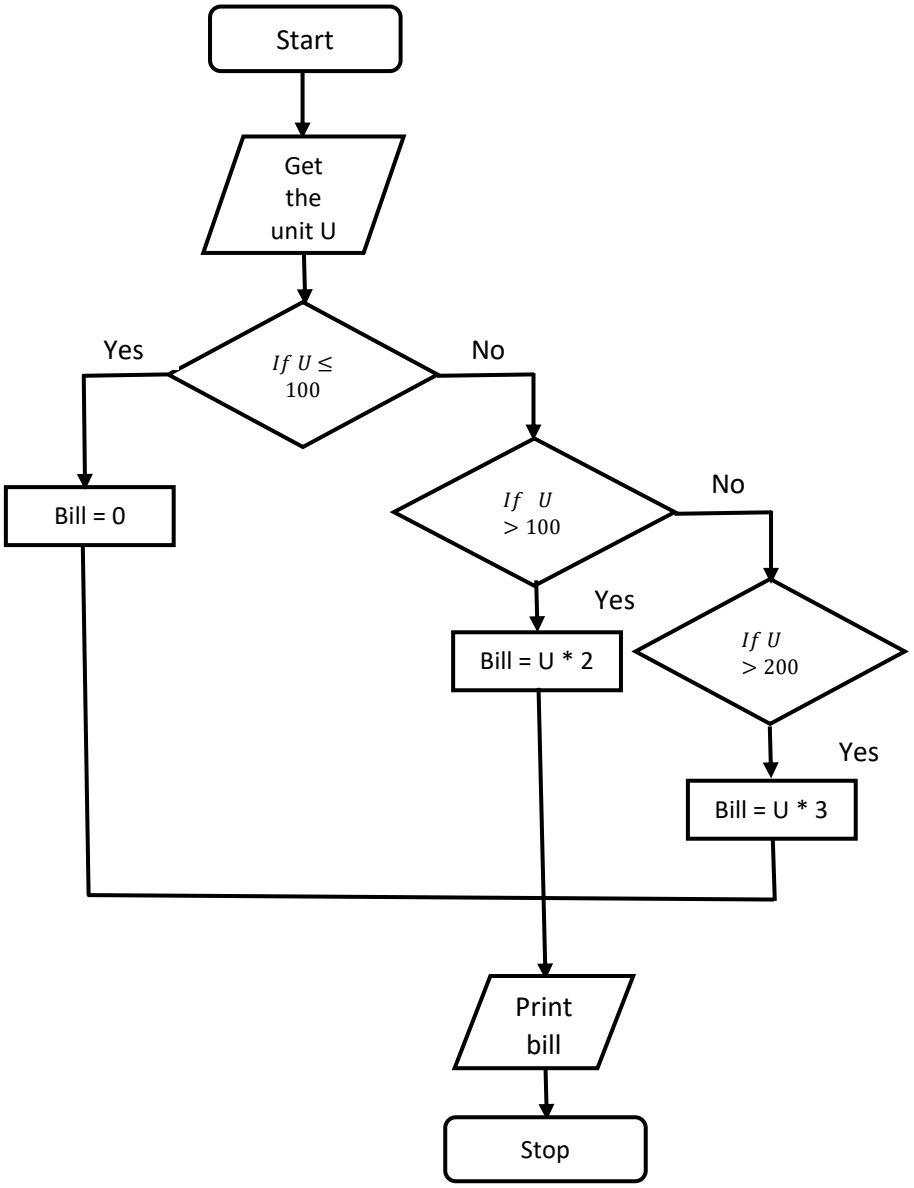
2. **Expected Learning outcomes :** It helps the students to understand to draw flowcharts for simple real time or scientific or technical problems.

3. **Flowcharts:**

A. Electricity Billing

Algorithm:

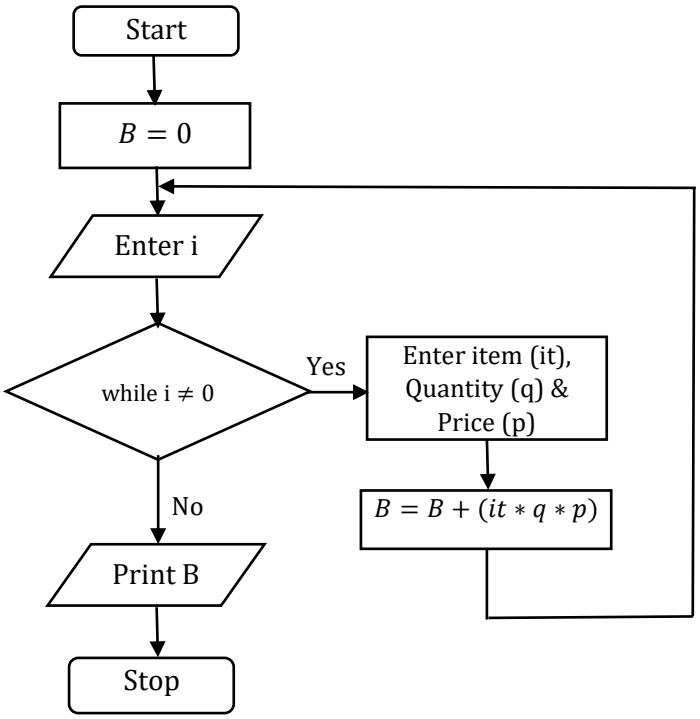
Step 1: Start the algorithm
Step 2: Get the number of units, U as input.
Step 3: i)For first 100 units , there is no bill.
ii)For units between 101 to 200, assign Rs 2 for an unit.
iii)for units greater than 200, calculate the bills by assigning Rs. 3 for each unit.
Step 4: Print the bill
Step 5: Stop the algorithm



B. Retail shop billing

Algorithm:

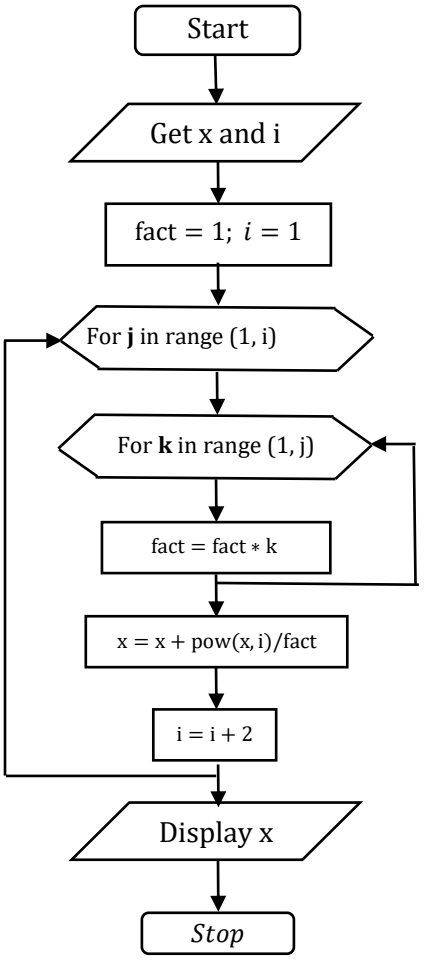
Step 1: Start the algorithm
Step 2: Enter option(i) as integer value
Step 3: If option(i) is not 0,
i)Get the item name(it),quantity(q) and its price(p) as input.
ii)Calculate the bill by it*q*p.
If the option(i) is equal to 0, come out of the loop
Step 4: Print the bill values.
Step 5: Stop the algorithm.



C. Sine Series:

Algorithm:

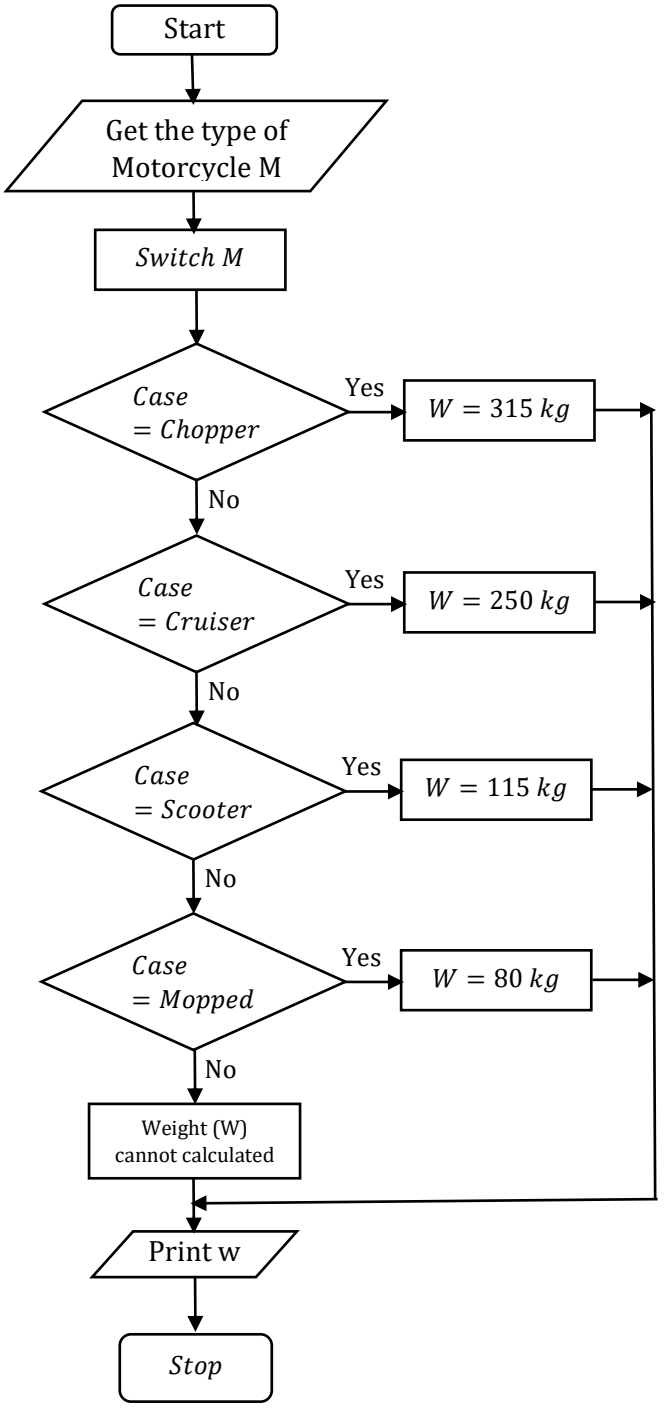
Step 1: Start the algorithm
Step 2: Get x and i
Step 3: Initialize fact=1 and i=1
Step 4: for k=1 to j
Calculate fact as fact * k
Find the factorial as x
Increment i by 2.
Step 5: Repeat step 4 for i times
Step 6: Display the sine series result
Step 7: Stop the algorithm.


Start
Get x and i
fact = 1; i = 1
For j in range (1, i)
For k in range (1, j)
fact = fact * k
x = x + pow(x, i)/fact
i = i + 2
Display x
Stop

D: weight of a motorbike

Algorithm:

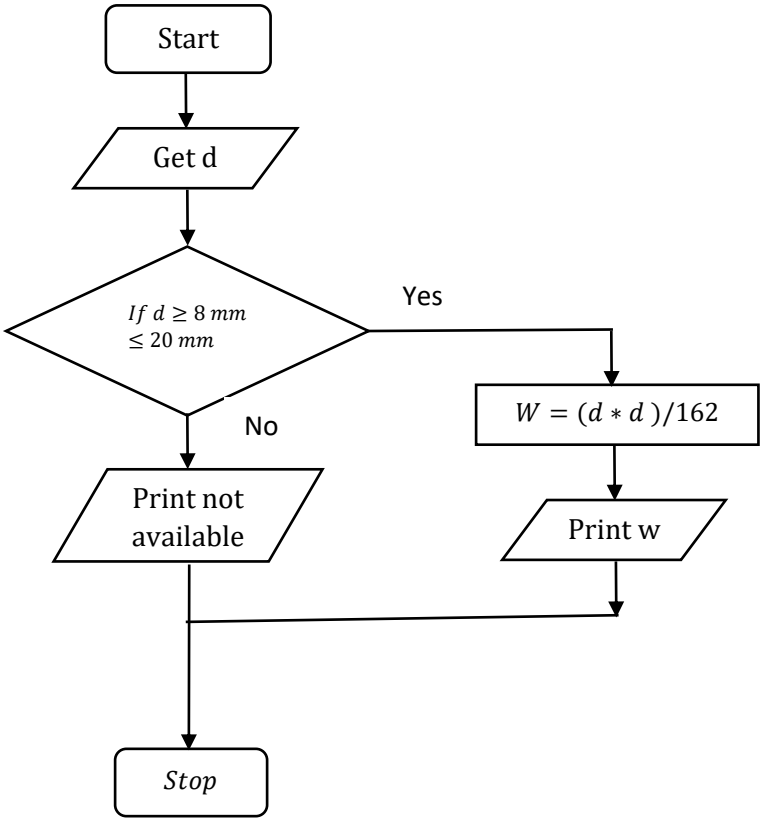
Step 1: Start the algorithm
Step 3: Get the types of motorcycle M
Step 4: Based on the type M, choose the weight as
If M=chopper, W= 315 kg
If M=Cruiser, W=250 kg
If M=Scooter, W=115 kg
If M=Mopped, W=80 kg
Else print as cannot find the weight
Step 5: Print the weight
Step 6: Stop the algorithm.



E. Weight of a steel bar

Algorithm:

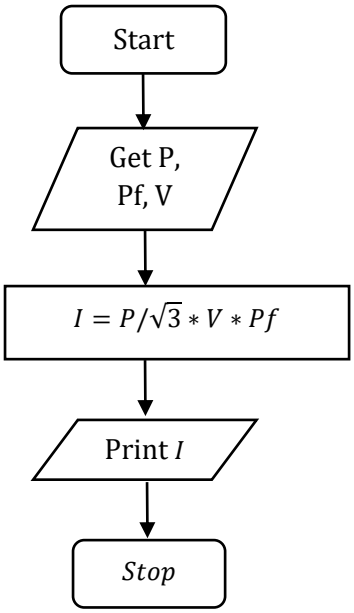
Step 1: Start the algorithm
Step 2: Get the diameter d of a steel bar.
Step 3: If d>= 8 mm and <= 20 mm then
Calculate the W as (d*d)/162
Print W
Else
Print as not available
Step 4: Stop the algorithm.



F. Compute Electrical Current in Three Phase AC Circuit

Algorithm:

Step 1: Start the algorithm
Step 2: Get the power factor Pf, Power P and Voltage V.
Step 3: Calculate the current I as $P/\sqrt{3} * V * Pf$
Step 4: Print the current I.
Step 5: Stop the algorithm



4. Result

Thus the flowcharts for Electricity Billing, Retail shop billing, Sine series, weight of a motorbike, Weight of a steel bar, compute Electrical Current in Three Phase AC Circuit are drawn.

5. Viva voice

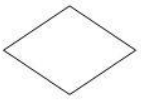
A. State the different ways to design a program

A program can be designed by using algorithm and flowcharts.

B. What is the difference between algorithm and flowcharts?

An algorithm is a step-by-step set of instructions for solving a problem in a limited number of steps; Flowchart- It is a diagrammatic representation of an algorithm.

C. List few symbols used in flowcharts.

	Start and end of a flowchart
	Processing of data or calculations
	Decision
	Input or output of data
	A document or report
	A subroutine
	A flow of logic

D. State the mechanism to develop an algorithm

Define the problem, List the inputs/outputs, describe the steps(start with high level and refine them), test with data.

E. List the basic building blocks of an algorithm.

The basic building blocks of an algorithms are: Instructions or statements, sequence, variables, conditional or selections statements, repetition or control flow structures, function or subroutine.

Ex.No: 2 **Problems Using simple statements and expressions**

Date :

1. **Problem Statement :** Write a python programs to exchange the values of two variables, circulate the values of n variables, distance between two points using simple statements and expressions.

2. **Expected Learning Outcomes :** It helps the students to understand the functionality of simple statements and expressions.

3. **Problem Analysis:**
   I. Using simple statement exchange the values of two variables should be implemented by getting two integer values x,y as input and print their values after swapping them.
   II. Using List, circulate the values of n variables by getting a value is to be circulated,list_1 and position from where it should circulate, n as inputs.
   III. Using simple statement and expressions, distance between two points are to be implemented. It gets two floating point values (x1,y1) and (x2,y2) and by applying the distance formula $\mathbf{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{½}}$ the distance between them is calculated and displayed.

4. A. **Expected Input:**

   I. enter x value : 50
   enter y value : 10

   II. enter n value :1

   III. enter x1 : 5
   enter y1 : 4
   enter x2 : 6
   enter y2 : 3

   B. **Expected Output :**

   I. After swapping value of x = 10

   After swapping value of y = 50

   II. Rotating [10, 20, 30, 40, 50] by 1 position is [50, 10, 20, 30, 40]

   III.Distance between points (5.0, 4.0) and (6.0, 3.0) is 1.4142135623730951

5. **Algorithm:**
   I.

Step 1: Start the program
Step 2 : Get the input for two variables x, y.
Step 3 : Have a third variable temp and first assign x to temp; then assign y value to x and
assign temp value to x
Step 4: Print the swapped value of x and y.
Step 5: Stop the program.

II.
Step 1 : Start the program
Step 2 : Initialize numbers using list.
Step 3 : Also get a value n for rotation.
Step 4 : Obtain the numbers available from the nth position, the numbers till n position and append the both values.
Step 5: Display the result.
Step 6: Stop the program.

III.

Step 1: Start the program.
Step 2: Get two points, namely(x1,y1) and (x2,y2).
Step 3: Calculate the distance between two points by using $((x_2 - x_1)^2 + (y_2 - y_1)^2)^{½}$
Step 4: Display the resultant distance value.
Step 5: Stop the program

6. Coding:

i) **Write a python program to exchange the values of two variables.**

```
# python program to swap two variables
#Get values for x and y
x= int(input("enter x value : "))
y= int(input("enter y value : "))
temp = x
x = y
y = temp
print("\nAfter swapping value of x =", x)
print("After swapping value of y =", y)
```

ii) Write a python program to circulate the values of n variable

```
# Right Rotating a list to n positions

n= int(input("enter n value : "))
list_1 = [1, 2, 3, 4, 5, 6]
list_2 = (list_1[-n:] + list_1[:-n])
print("Rotating {} by {} position is {}".format(list_1,n,list_2))
```

iii) Write a python program to calculate the distance between two points.

```
#Python program to find distance between two points
# point a
x1 = float(input(“enter x1 : ”))
y1 = float(input(“enter y1 : ”))
# point b
x2 = float(input(“enter x2 : ”))
y2 = float(input(“enter y2 : ”))

# distance between a and b
distance = ((x1 - x2)**2 + (y1 - y2)**2)**0.5
# display the result
print("Distance between points ({}, {}) and ({}, {}) is
{}".format(x1,y1,x2,y2,distance))
```

7. **Test cases:**

	Input	**Output**
I	enter x value : 45 enter y value : 34	After swapping value of x = 34 After swapping value of y = 45
	enter x value : 4.5	ValueError: invalid literal for int() with base 10: '4.5'
	enter x value : g	NameError: name 'g' is not defined
	enter x value : '33'	ValueError: invalid literal for int() with base 10: "'33'"
II	enter n value :1 enter n value :2 enter n value : 3	Rotating [1, 2, 3, 4, 5, 6] by 1 position is [6, 1, 2, 3, 4, 5] Rotating [1, 2, 3, 4, 5, 6] by 2 position is [5, 6, 1, 2, 3, 4] Rotating [1, 2, 3, 4, 5, 6] by 3 position is [4, 5, 6, 1, 2, 3]
	enter n value : 2	Rotating [1.0, 2.6, 3.5, 4.8, 5.9, 6.3] by 2 position is [5.9, 6.3, 1.0, 2.6, 3.5, 4.8]
	enter n value : 2	Rotating ['a', 'b', 'c', 'd', 'e'] by 2 position is ['d', 'e', 'a', 'b', 'c']
	enter n value : 3	Rotating ['champa', 'lotus', 'lily', 'rose', 'jasmine'] by 3 position is ['lily', 'rose', 'jasmine', 'champa', 'lotus']
III	enter x1 : 2 enter y1 : 3 enter x2 : 5 enter y2 : 7	Distance between points (2.0, 3.0) and (5.0, 7.0) is 5.0
	enter x1 : a	name 'a' is not defined
	enter x1 : 'a'	ValueError: could not convert string to float: "'a'"

8. Output:

i)enter x value : 77
enter y value : 105
After swapping value of x = 105
After swapping value of y = 77

ii)enter n value : 4
Rotating [45, 76, 34, 65, 34, 90, 23] by 4 position is [65, 34, 90, 23, 45, 76, 34]

iii)enter x1 : 56
enter y1 : 4
enter x2 : 51
enter y2 : 4
Distance between points (56.0, 4.0) and (51.0, 4.0) is 5.0

9. Result

Thus the python program to swap the values of two numbers, circulate the values of n variables, distance between two points using simple statements and expressions are executed successfully.

10. Viva voice

A. **What are the data types supported by python?**

The data types supported by python are :
Binary Types: memoryview, bytearray, bytes.
Boolean Type: bool.
Set Types: frozenset, set.
Mapping Type: dict.
Sequence Types: range, tuple, list.
Numeric Types: complex, float, int.
Text Type: str.

B. **Define algorithm.**

An algorithm is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

C. **Difference between int and float datatype.**

Integer**s** and floats are two different kinds of numerical data. An integer (more commonly called an int) is a number without a decimal point. A float is a floating-point number, which means it is a number that has a decimal place. Floats are used when more precision is needed.

D. **What is a string?**

A string is a data type used in programming to represent text rather than **numbers**. It is comprised of a set of characters that can also contain spaces and numbers. For example, the word "hamburger" and the phrase "I ate 3 hamburgers" are both strings.

E. **Give an example for string data type?**
“hello”, “4 apple”

f) **What is meant by List in python?**

Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage. Lists are created using square brackets.

```
mylist = ["apple", "banana", "cherry"]
```

Ex.No: 3 Problems Using Conditionals and Iterative Loops

Date :

1. **Problem Statement**: Write a python program to create a sequence of number formed by various number series, number patterns and pyramid pattern

2. **Expected Learning Outcomes**: Train the students to understand the basics of various conditional and looping statements

3. **Problem Analysis**:

   I. Identify the appropriate looping statement to implement the number series which gets the N value from the user and prints sum of first N value and squares of first N values.

   II. Using appropriate conditional and looping statement implement a number patterns which gets a two number's X and Y. X and Y should be an positive number and X should be greater than Y (X>Y). Keep decrementing X from Y until you encounter 0. Once it reaches 0 keep incrementing X by Y until the value reaches X or above

   III. Using loops construct a pyramid. Generate the pyramid with an alphabet patterns or star patterns. Get the number of rows R in the pyramid from the user.

4. **A. Expected Input:**

   I. Please enter the value of N:6

   II. Please enter the value of X : 13
   Please enter the value of Y: 5

   III. Please the Number of rows R: 7

   B. Expected Output:

   I. The Square of the given number 6 is:
   1 5 14 30 55 91
   The Sum of the given number 6 is:
   1 3 6 10 15 21

   II. The Series for 13 and 5 is: 13 8 3 2 7 12 17

III.

```
      *
    * *
   * * *
  * * * *
 * * * * *
* * * * * *
* * * * * * *
```

5. Algorithm:

I.

Step 1: Get the input from the user and store the value in n
Step 2: Initialize two variables to store the square and sum
Step 3: Initialize i=1
Step 4: Repeat Step 5 through Step 7 until i <= n
Step 5: Calculate Square = Square + (i*i)
Step 6: Calculate Sum = Sum + i
Step 7: Increment i by 1
Step 8 : Display Square and Sum
Step 9: Stop

II.

Step 1: Get the X and Y values as input from the user
Step 2: Initialize two variables to store the square and sum
Step 3: Initialize i=1
Step 4: Repeat Step 5 through Step 12 until i <= X*2+Y
Step 5: if X-i >= 0 goto Step 6 else goto Step 8
Step 6: Calculate a = X-i
Step 7 : Display a
Step 8: if X-i < 0 goto Step 9 else goto Step 12
Step 9: Calculate a = X-i
Step 10: Calculate a = a + a*-2
Step 11: Display a
Step 12: Increment i by 1
Step 13: Stop

III.

Step 1: Get the number of rows R from the user
Step 2: Initialize a temporary variable K to count on number of spaces
Step 3: Initialize i=1. Create an outer loop to handle number of rows.
Step 4: Repeat Step 5 through Step 8 until i <=n
Step 5: Initialize j=1. Create inner loop to handle number spaces.
Step 6: Repeat Step 7 through Step 8 until j <=n
Step 7: Display a space
Step 8: Decrement K by 1 after each loop
Step 9: Initialize j=1. Create inner loop to handle number of columns.
Step 10: Repeat Step 11 through Step 13 until j <=n
Step 11: Display '*'
Step 12: Increment i
Step 13: Stop

6. Coding:

I. Write a program to print Square series and addition series. Display square of N natural numbers and sum of N natural numbers

```
N = int(input("Please enter the value of N:"))

Square = 0
Sum = 0
print('The Square of the given number',N,'is: ')
for i in range(1, N+1) :
  Square  = Square + (i * i)
  print(Square,end=' ')
print('\n',end=' ')
print('The Sum of the given number',N,'is: ')
for i in range(1, N+1) :
  Sum = Sum + i
  print(Sum,end=' ')
```

II. Write a program to display number pattern

```
X = int(input("Please enter the value of X:"))
Y = int(input("Please enter the value of Y:"))
print('The Series for',X,'and',Y,'is:',end=' ' )
for i in range(0,X*2+Y,Y):
    if X-i >= 0:
        a = X-i
        print(a,end=' ')
    elif X-i < 0:
        a = X-i
        a = a + a*-2
        print(a,end=' ')
```

III. Write a program to display pyramid pattern of N srows

```
n = int(input("Please enter the value of N:"))
k = n - 1

for i in range(0, n):
  for j in range(0, k):
    print(end=" ")
  k = k - 1
  for j in range(0, i+1):
    print("* ", end="")
  print("\r")
```

7. Test case :

	Input	Output
I	Please enter the value of N:10	The Square of the given number 10 is: 1 5 14 30 55 91 140 204 285 385 The Sum of the given number 10 is: 1 3 6 10 15 21 28 36 45 55
	Please enter the value of N:-2	The Square of the given number -2 is: The Sum of the given number -2 is:
	Please enter the value of N:a	invalid literal for int() with base 10: 'a'
	Please enter the value of N:'1'	invalid literal for int() with base 10: "'1'"
II	Please enter the value of X:14 Please enter the value of Y:3	The Series for 14 and 3 is: 14 11 8 5 2 1 4 7 10 13 16
	Please enter the value of X:12 Please enter the value of Y:0	range() arg 3 must not be zero
III	Please the Number of rows R:-1	**No output**
	Please the Number of rows R:1	*
	Please the Number of rows R:03	* * * * * *

8. Output :

i. The Square of the given number 6 is: 91
The Sum of the given number 6 is: 21

ii. The Series for 13 and 5 is: 13 8 3 2 7 12 17

iii.

```
     *
    * *
   * * *
  * * * *
 * * * * *
* * * * * *
* * * * * * *
```

9. Result

Number series, Number pattern and pyramid pattern were successfully implemented through conditional if statement, nested if statement, looping statement through for loop and nested for loop

10. Viva Questions

A. Which operator is used to check the equality of the two operands?

The equal-to operator (==) returns true if both operands have the same value; otherwise, it returns false .

B. Do we require a Colon at the end of the conditional and looping statement . Justify if required.

The colon (:) is significant and required. It separates the header of the compound statement from the body.

The line after the colon must be indented. It is standard in Python to use four spaces for indenting.

All lines indented the same amount after the colon will be executed whenever the BOOLEAN_EXPRESSION is true.

C. Is it necessary to use else statement whenever if statement are used

No it is not required to have else block whenever we use if block

D. Do all the statement of the block has the same indent.

Yes. One of the distinctive features of Python is its use of indentation to highlighting the blocks of code. Whitespace is used for indentation in Python. All statements with the same distance to the right belong to the same block of code.

E. **What will be the sequence generated by range(10) statement?**

[0,1,2,3,4,5,6,7,8,9]

Ex.No: 4 Implementing real-time/technical applications using Lists, Tuples.

Date :

1. **Problem Statement**: Write a python program to create and perform the operations of list and tuple for the items present in a library

2. **Expected Learning Outcomes**: Train the students to understand the operations of list and tuple

3. **Problem Analysis**:

   I. Using list, implement operations of addition and deletion of items in library, and differentiate the operations which can be performed by tuple.
   II. Perform the slicing operations for the materials required for construction using list and tuple

4. **A. Expected Input:**

   I. Enter the items in the library separated by space: Books Journals Articles Novels
   II. Enter the Material List required for construction separated by space: mSand Steel Stones PVC Plaster Putty
   III. Enter the Material Tuple required for construction separated by space:Cement Sand Bars Bricks Paint Wood Tiles

   B. Expected Output:

   I.

Enter the items in the library separated by space: Books Journals Articles Novels
List of items in the library: ['Books', 'Journals', 'Articles', 'Novels']
List of items in the library after appending new item: ['Books', 'Journals', 'Articles', 'Novels', 'Newspaper']
List of items in the library after adding new item at 3rd index: ['Books', 'Journals', 'Articles', 'eAssets', 'Novels', 'Newspaper']
List of items in the library after removing an item: ['Books', 'Journals', 'Articles', 'Novels', 'Newspaper']
List of items in the library after removing an item at 2nd index: ['Books', 'Journals', 'Novels', 'Newspaper']
The list library is converted to tuple library now
The items in the tuple library is ('Books', 'Journals', 'Novels', 'Newspaper')
Length of the tuple library is: 4
Yes, 'Newspaper' is an item in the library tuple

II.

Actual string: Common Materials required for construction
The substring after slice (7, 25): Materials required
Original List: ['mSand', 'Steel', 'Stones', 'PVC', 'Plaster', 'Putty']
Sliced List with slice(1, 5, 1): ['Steel', 'Stones', 'PVC', 'Plaster']
The List after negative values for slice: ['Putty', 'Plaster', 'PVC']
Sliced List with [2:6:1: ['Stones', 'PVC', 'Plaster', 'Putty']
Sliced List with [1::]: ['Steel', 'Stones', 'PVC', 'Plaster', 'Putty']
Original Tuple: ['Cement', 'Sand', 'Bars', 'Bricks', 'Paint', 'Wood', 'Tiles']
The tuple after slice: ['Cement', 'Sand', 'Bars', 'Bricks']
Sliced Tuple with [:]: ['Cement', 'Sand', 'Bars', 'Bricks', 'Paint', 'Wood', 'Tiles']
Sliced Tuple with [::-1] = ['Tiles', 'Wood', 'Paint', 'Bricks', 'Bars', 'Sand', 'Cement']
Sliced Tuple with [-1:-4:-1] = ['Tiles', 'Wood', 'Paint']

5. Algorithm:

I.

Step 1: Get the items of library as input from the user
Step 2: Store the inputs in a list.
Step 3: Append a new item in the list.
Step 4: Add a new item to 3rd index of the list
Step 5: Remove an existing item from the list
Step 6: Remove an existing item from 2nd index of the list
Step 7: Convert the list into tuple
Step 8: Identify an item in the tuple by If condition
Step 9: Stop

II.

Step 1: Print the actual string
Step 2: Slice and print the substring from the string
Step 3: Get the common materials required for construction as input from the user as List
Step 4: Store the inputs in a List
Step 5: Perform slicing operations in the list
Step 6: Get the common materials required for construction as input from the user as Tuple
Step 7: Store the inputs in a Tuple
Step 8: Perform slicing operations in the Tuple
Step 9: Stop

6. Coding:

I.

Write a python program to create and perform the operations of list and tuple for the items present in a library

```
library = list(map(str, input("Enter the items in the library separated by space: ").split()))
print("List of items in the library:",library)
library.append("Newspaper")
print("List of items in the library after appending new item:",library)
library.insert(3,"eAssets")
print("List of items in the library after adding new item at 3rd index:",library)
library.remove("eAssets")
print("List of items in the library after removing an item:",library)
del library[2]
print("List of items in the library after removing an item at 2nd index:",library)
print("The list library is converted to tuple library now")
libraryt = tuple(library)
print("The items in the tuple library is",libraryt)
print("Length of the tuple library is:",len(libraryt))
if "Newspaper" in libraryt:
   print("Yes, 'Newspaper' is an item in the library tuple")
else:
   print("No, 'Newspaper' is not an item in the library tuple")
```

II.

Write a python program to perform the slicing operations of list and tuple for the materials required for construction

```
str_x = "Common Materials required for construction"
print("Actual string: ", str_x)
obj_slice = slice(7, 25)
print("The substring after slice (7, 25):" ,str_x[obj_slice])
MaterialList = list(map(str, input("Enter the Material List required for construction separated by space: ").split()))
print("Original List:", MaterialList)
obj_slice = slice(1, 5, 1)
print("Sliced List with slice(1, 5, 1):", MaterialList[obj_slice])
obj_slice = slice(-1,-4, -1)
print("The List after negative values for slice:" ,MaterialList[obj_slice])
print("Sliced List with [2:6:1:", MaterialList[2:6:1])
print("Sliced List with [1::]:", MaterialList[1::])
MaterialTuple = list(map(str, input("Enter the Material Tuple required for construction separated by space:").split()))
```

```
print("Original Tuple:", MaterialTuple)
obj_slice = slice(4)
print("The tuple after slice:" ,MaterialTuple[obj_slice])
print("Sliced Tuple with [:]:", MaterialTuple[:])
print("Sliced Tuple with [::-1] = ", MaterialTuple[::-1])
print("Sliced Tuple with [-1:-4:-1] = ", MaterialTuple[-1:-4:-1])
```

7. Test case :

	Input	**Output**
I	Enter the items in the library separated by space: Books Journals Articles Novels	Enter the items in the library separated by space: Books Journals Articles Novels List of items in the library: ['Books', 'Journals', 'Articles', 'Novels'] List of items in the library after appending new item: ['Books', 'Journals', 'Articles', 'Novels', 'Newspaper'] List of items in the library after adding new item at 3rd index: ['Books', 'Journals', 'Articles', 'eAssets', 'Novels', 'Newspaper'] List of items in the library after removing an item: ['Books', 'Journals', 'Articles', 'Novels', 'Newspaper'] List of items in the library after removing an item at 2nd index: ['Books', 'Journals', 'Novels', 'Newspaper'] The list library is converted to tuple library now The items in the tuple library is ('Books', 'Journals', 'Novels', 'Newspaper') Length of the tuple library is: 4
	Enter the items in the library separated by space: Books,Journals,Articles,Novels	IndexError: list assignment index out of range
	Enter the items in the library separated by space: 1 2 3 4	Enter the items in the library separated by space: 1 2 3 4 List of items in the library: ['1', '2', '3', '4'] List of items in the library after appending new item: ['1', '2', '3', '4', 'Newspaper'] List of items in the library after adding new item at 3rd index: ['1', '2', '3', 'eAssets', '4', 'Newspaper'] List of items in the library after removing an item: ['1', '2', '3', '4', 'Newspaper'] List of items in the library after removing an item at 2nd index: ['1', '2', '4', 'Newspaper'] The list library is converted to tuple library now The items in the tuple library is ('1', '2', '4', 'Newspaper') Length of the tuple library is: 4 Yes, 'Newspaper' is an item in the library tuple
II	Enter the Material List required for construction separated by space: mSand Steel Stones PVC Plaster Putty Enter the Material Tuple required for construction separated by space:Cement Sand Bars Bricks Paint Wood Tiles	Actual string: Common Materials required for construction The substring after slice (7, 25): Materials required Original List: ['mSand', 'Steel', 'Stones', 'PVC', 'Plaster', 'Putty'] Sliced List with slice(1, 5, 1): ['Steel', 'Stones', 'PVC', 'Plaster'] The List after negative values for slice: ['Putty',

		'Plaster', 'PVC'] Sliced List with [2:6:1: ['Stones', 'PVC', 'Plaster', 'Putty'] Sliced List with [1::]: ['Steel', 'Stones', 'PVC', 'Plaster', 'Putty'] Original Tuple: ['Cement', 'Sand', 'Bars', 'Bricks', 'Paint', 'Wood', 'Tiles'] The tuple after slice: ['Cement', 'Sand', 'Bars', 'Bricks'] Sliced Tuple with [:]: ['Cement', 'Sand', 'Bars', 'Bricks', 'Paint', 'Wood', 'Tiles'] Sliced Tuple with [::-1] = ['Tiles', 'Wood', 'Paint', 'Bricks', 'Bars', 'Sand', 'Cement'] Sliced Tuple with [-1:-4:-1] = ['Tiles', 'Wood', 'Paint']
	Enter the Material List required for construction separated by space: 1 2 3 4 5 6 Enter the Material Tuple required for construction separated by space:1 2 3 4 5 6	Actual string: Common Materials required for construction The substring after slice (7, 25): Materials required Original List: ['1', '2', '3', '4', '5', '6'] Sliced List with slice(1, 5, 1): ['2', '3', '4', '5'] The List after negative values for slice: ['6', '5', '4'] Sliced List with [2:6:1: ['3', '4', '5', '6'] Sliced List with [1::]: ['2', '3', '4', '5', '6'] Original Tuple: ['1', '2', '3', '4', '5', '6'] The tuple after slice: ['1', '2', '3', '4'] Sliced Tuple with [:]: ['1', '2', '3', '4', '5', '6'] Sliced Tuple with [::-1] = ['6', '5', '4', '3', '2', '1'] Sliced Tuple with [-1:-4:-1] = ['6', '5', '4']

8. Output :

I.

Enter the items in the library separated by space: Books Journals Articles Novels
List of items in the library: ['Books', 'Journals', 'Articles', 'Novels']
List of items in the library after appending new item: ['Books', 'Journals', 'Articles', 'Novels', 'Newspaper']
List of items in the library after adding new item at 3rd index: ['Books', 'Journals', 'Articles', 'eAssets', 'Novels', 'Newspaper']
List of items in the library after removing an item: ['Books', 'Journals', 'Articles', 'Novels', 'Newspaper']
List of items in the library after removing an item at 2nd index: ['Books', 'Journals', 'Novels', 'Newspaper']
The list library is converted to tuple library now
The items in the tuple library is ('Books', 'Journals', 'Novels', 'Newspaper')
Length of the tuple library is: 4
Yes, 'Newspaper' is an item in the library tuple

II.

Actual string: Common Materials required for construction
The substring after slice (7, 25): Materials required
Original List: ['mSand', 'Steel', 'Stones', 'PVC', 'Plaster', 'Putty']
Sliced List with slice(1, 5, 1): ['Steel', 'Stones', 'PVC', 'Plaster']

The List after negative values for slice: ['Putty', 'Plaster', 'PVC']
Sliced List with [2:6:1: ['Stones', 'PVC', 'Plaster', 'Putty']
Sliced List with [1::]: ['Steel', 'Stones', 'PVC', 'Plaster', 'Putty']
Original Tuple: ['Cement', 'Sand', 'Bars', 'Bricks', 'Paint', 'Wood', 'Tiles']
The tuple after slice: ['Cement', 'Sand', 'Bars', 'Bricks']
Sliced Tuple with [:]: ['Cement', 'Sand', 'Bars', 'Bricks', 'Paint', 'Wood', 'Tiles']
Sliced Tuple with [::-1] = ['Tiles', 'Wood', 'Paint', 'Bricks', 'Bars', 'Sand', 'Cement']
Sliced Tuple with [-1:-4:-1] = ['Tiles', 'Wood', 'Paint']

9. Result

The operations of list and tuple were successfully implemented for the items in library.

10. Viva Questions

F. How do you represent a list and tuple?

List can be represented by [ ]
Tuple can be represented by ( )

G. What operations can and cannot be performed in tuple.

We can't add elements to a tuple. There's no append() or extend() method for tuples.
We can't remove elements from a tuple. Tuples have no remove() or pop() method.
We can find elements in a tuple since this doesn't change the tuple.
We can also use the in operator to check if an element exists in the tuple.

H. Are lists hashable ?

No lists are not hashable, whereas tuple is hashable

I. How to concatenate lists in Python.

To concatenate lists, we use the + operator. This will give us a new list that is the concatenation of our two lists without modifying the original ones.

J. How to return tuples as values?

A function can only return one value, but if the value is a tuple, the effect is the same as returning multiple values.

K. How do you reverse the order of sequence for slicing?

By using the negative value, we may reverse the order of sequence for slicing. If we want to count from the last item and return complete list, tuple etc., then use the -1 value.

Ex.No: 5 Implementing real-time/technical applications using Sets, Dictionaries.

Date :

1. **Problem Statement:** Write python programs to perform the various operations using Sets and Dictionaries.

2. **Expected Learning Outcomes**: Train the students to understand the operations available for Sets and Dictionaries in python using real-time examples

3. Problem Analysis:
   - I. Use the possible methods to perform the create, access, add, length, remove, clear and delete operations of Language sets in python.
   - II. Using dictionaries perform the create, access, change, add, remove, clear and delete operations of automobile components in python.
   - III. Perform the conditional check, iteration, length, and sort operations for the dictionary with the elements of civil structure.

4. **A. Expected Input:**

   I.

   Enter the Languages separated by space: English French German

   II.

   Enter input pair for Key 'Chasis' : Hatchback
   Enter input pair for Key 'Engine' : VCylinder
   Enter input pair for Key 'Capacity' : 1000cc
   Enter input pair for Key 'Variant' : Petrol

   III.

   Enter number of 'Pillars': 100
   Enter number of 'Floors': 5
   Enter number of 'Windows': 50

   B. Expected Output:

   I.

   {'French', 'German', 'English'}
   True
   {'French', 'Chinese', 'German', 'English'}
   {'French', 'Malayalam', 'Tamil', 'Telugu', 'German', 'Chinese', 'English'}
   7
   {'French', 'Malayalam', 'Tamil', 'Telugu', 'German', 'English'}
   {'Malayalam', 'Tamil', 'Telugu', 'German', 'English'}

```
Malayalam
{'Tamil', 'Telugu', 'German', 'English'}
set()
Traceback (most recent call last):
  File "<string>", line 31, in <module>
NameError: name 'Language' is not defined
```

II.

```
{'Chasis': 'Hatchback', 'Engine': 'VCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol'}
VCylinder
VCylinder
{'Chasis': 'Hatchback', 'Engine': 'VCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol',
'Cost': '75000'}
{'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc', 'Variant':
'Petrol', 'Cost': '75000'}
75000
{'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc', 'Variant':
'Petrol'}
('Variant', 'Petrol')
{'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc'}
{'Chasis': 'Hatchback', 'Capacity': '1000cc'}
{}
Traceback (most recent call last):
  File "<string>", line 35, in <module>
NameError: name 'automobile' is not defined
```

III.

```
{'Pillars': '100', 'Floors': '5', 'Windows': '50'}
Contents of the Dictionary = {'Pillars': '100', 'Floors': '5', 'Windows': '50'}
Checking if Pillars is a key in dictionary = True
Checking if Walls is not a key in dictionary = True
Cannot check with pair value 50 in dictionary (True - pair can be checked, False -
pair cannot be checked = False
Pair values in the dictionary =
100
5
50
Length of the dictionary = 3
Sort the keys of the dictionary in alphabetical order = ['Floors', 'Pillars', 'Windows']
```

5. Algorithm:

I.

Step 1: Create a language set with user inputs as English, French and German
Step 2: Check/Access whether German is in the set.
Step 3: Add Chinese to the set
Step 4: Update the set with Tamil, Telugu and Malayalam
Step 5: Print the length of the set
Step 6: Remove Chinese from the set
Step 7: Discard French from the set

Step 8: Use Pop to do random removal of the set
Step 9: Clear the set
Step 10: Delete the set
Step 11: Display the error on printing deleted set
Step 12: Stop

II.

Step 1: Create an automobile component dictionary pair, key: value as Type: Bike, Engine: 2Stroke, Capacity: 100cc}
Step 2: Access the value for the Key "Engine"
Step 3: Access the value for the Key "Engine" using get
Step 4: Add a new key: value as Cost: 75000
Step 5: Update the pair for the key "Engine" as "4Stroke"
Step 6: Remove the particular "Cost" key using pop
Step 7: Remove an arbitrary key using popitem
Step 8: Delete the particular "Engine" key
Step 9: Clear the dictionary
Step 10: Delete the dictionary
Step 11: Display the error on printing deleted dictionary
Step 12: Stop

III.

Step 1: Create an civil elements dictionary pair, key: value as Pillars: 100, Floors: 5, Windows: 50, Doors: 60
Step 2: List the contents of the dictionary
Step 3: Check if "Pillars" is an element key
Step 4: Check if "Walls" is not an element key
Step 5: Check if pair value can be checked in dictionary
Step 6: List the pair values using for loop iteration
Step 7: Display the length of the dictionary
Step 8: Sort the element keys in alphabetical order
Step 9: Stop

6. Coding:

I. Write a python program to perform the create, access, add, length, remove, clear and delete operations of Language sets.

```
#Step 1
list = list(map(str, input("Enter the Languages separated by space: ").split()))
Language = set(list)
print(Language)
#Step 2
print("German" in Language)
#Step 3
Language.add("Chinese")
print(Language)
```

```
#Step 4
Language.update(["Tamil", "Telugu", "Malayalam"])
print(Language)
#Step 5
print(len(Language))
#Step 6
Language.remove("Chinese")
print(Language)
#Step 7
Language.discard("French")
print(Language)
#Step 8
x = Language.pop()
print(x)
print(Language)
#Step 9
Language.clear()
print(Language)
#Step 10
del Language
#Step 11
print(Language)
```

II. Write a python program to perform the create, access, change, add, length, remove, clear and delete operations of automobile component dictionary.

```
#Step 1
v1=input("Enter input pair for Key 'Chasis' : ")
v2=input("Enter input pair for Key 'Engine' : ")
v3=input("Enter input pair for Key 'Capacity' : ")
v4=input("Enter input pair for Key 'Variant' : ")
automobile={"Chasis":v1, "Engine":v2, "Capacity":v3, "Variant":v4}
print(automobile)
#Step 2
x = automobile["Engine"]
print(x)
#Step 3
x = automobile.get("Engine")
print(x)
#Step 4
automobile['Cost'] = '75000'
print(automobile)
#Step 5
automobile['Engine'] = "TwinCylinder"
print(automobile)
#Step 6
print(automobile.pop("Cost"))
```

```
print(automobile)
#Step 7
print(automobile.popitem())
print(automobile)
#Step 8
del automobile["Engine"]
print(automobile)
#Step 9
automobile.clear()
print(automobile)
#Step 10
del automobile
#Step 11
print(automobile)
```

III. Write a python program to perform the conditional check, iteration, length, and sort operations for the dictionary with the elements of civil structure.

```
v1=input("Enter number of 'Pillars': ")
v2=input("Enter number of 'Floors': ")
v3=input("Enter number of 'Windows': ")
civilelements={"Pillars":v1, "Floors":v2, "Windows":v3}
print(civilelements)
print("Contents of the Dictionary =", civilelements)
print("Checking if Pillars is a key in dictionary =", "Pillars" in civilelements)
print("Checking if Walls is not a key in dictionary =", "Walls" not in civilelements)
print("Cannot check with pair value 50 in dictionary (True - pair can be checked, False - pair cannot be checked =", "50" in civilelements)
print("Pair values in the dictionary =")
for i in civilelements:
 print(civilelements[i])
print("Length of the dictionary =", len(civilelements))
print("Sort the keys of the dictionary in alphabetical order =", sorted(civilelements))
```

7. Test case:

	Input	Output
I.	Enter the Languages separated by space: English French German	{'French', 'German', 'English'} True {'French', 'Chinese', 'German', 'English'} {'French', 'Malayalam', 'Tamil', 'Telugu', 'German', 'Chinese', 'English'} 7 {'French', 'Malayalam', 'Tamil', 'Telugu', 'German', 'English'} {'Malayalam', 'Tamil', 'Telugu', 'German', 'English'} Malayalam {'Tamil', 'Telugu', 'German', 'English'} set() Traceback (most recent call last): File "<string>", line 31, in <module>

		NameError: name 'Language' is not defined
	Enter the Languages separated by space: English, French, German	{'German', 'English,', 'French,'} True {'Chinese', 'German', 'English,', 'French,'} {'German', 'Telugu', 'Chinese', 'Tamil', 'English,', 'Malayalam', 'French,'} 7 {'German', 'Telugu', 'Tamil', 'English,', 'Malayalam', 'French,'} {'German', 'Telugu', 'Tamil', 'English,', 'Malayalam', 'French,'} German {'Telugu', 'Tamil', 'English,', 'Malayalam', 'French,'} set() Traceback (most recent call last): File "<string>", line 31, in <module> NameError: name 'Language' is not defined
II	Enter input pair for Key 'Chasis' : Hatchback Enter input pair for Key 'Engine' : VCylinder Enter input pair for Key 'Capacity' : 1000cc Enter input pair for Key 'Variant' : Petrol	{'Chasis': 'Hatchback', 'Engine': 'VCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol'} VCylinder VCylinder {'Chasis': 'Hatchback', 'Engine': 'VCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol', 'Cost': '75000'} {'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol', 'Cost': '75000'} 75000 {'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol'} ('Variant', 'Petrol') {'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc'} {'Chasis': 'Hatchback', 'Capacity': '1000cc'} {} Traceback (most recent call last): File "<string>", line 35, in <module> NameError: name 'automobile' is not defined
	Enter input pair for Key 'Chasis' : 1 Enter input pair for Key 'Engine' : 2 Enter input pair for Key 'Capacity' : 3	{'Chasis': '1', 'Engine': '2', 'Capacity': '3', 'Variant': '4'} 2 2 {'Chasis': '1', 'Engine': '2', 'Capacity': '3', 'Variant': '4', 'Cost': '75000'} {'Chasis': '1', 'Engine': 'TwinCylinder', 'Capacity': '3', 'Variant': '4', 'Cost': '75000'} 75000 {'Chasis': '1', 'Engine': 'TwinCylinder', 'Capacity': '3', 'Variant': '4'} ('Variant', '4') {'Chasis': '1', 'Engine': 'TwinCylinder', 'Capacity': '3'}

	Enter input pair for Key 'Variant' : 4	{'Chasis': '1', 'Capacity': '3'} {} Traceback (most recent call last): File "<string>", line 35, in <module> NameError: name 'automobile' is not defined
III	Enter number of 'Pillars': 100 Enter number of 'Floors': 5 Enter number of 'Windows': 50	{'Pillars': '100', 'Floors': '5', 'Windows': '50'} Contents of the Dictionary = {'Pillars': '100', 'Floors': '5', 'Windows': '50'} Checking if Pillars is a key in dictionary = True Checking if Walls is not a key in dictionary = True Cannot check with pair value 50 in dictionary (True - pair can be checked, False - pair cannot be checked = False Pair values in the dictionary = 100 5 50 Length of the dictionary = 3 Sort the keys of the dictionary in alphabetical order = ['Floors', 'Pillars', 'Windows']
	Enter number of 'Pillars': a Enter number of 'Floors': b Enter number of 'Windows': c	{'Pillars': 'a', 'Floors': 'b', 'Windows': 'c'} Contents of the Dictionary = {'Pillars': 'a', 'Floors': 'b', 'Windows': 'c'} Checking if Pillars is a key in dictionary = True Checking if Walls is not a key in dictionary = True Cannot check with pair value 50 in dictionary (True - pair can be checked, False - pair cannot be checked = False Pair values in the dictionary = a b c Length of the dictionary = 3 Sort the keys of the dictionary in alphabetical order = ['Floors', 'Pillars', 'Windows']

8. Output

I.

{'German', 'English,', 'French,'}
True
{'Chinese', 'German', 'English,', 'French,'}
{'German', 'Telugu', 'Chinese', 'Tamil', 'English,', 'Malayalam', 'French,'}
7
{'German', 'Telugu', 'Tamil', 'English,', 'Malayalam', 'French,'}
{'German', 'Telugu', 'Tamil', 'English,', 'Malayalam', 'French,'}
German
{'Telugu', 'Tamil', 'English,', 'Malayalam', 'French,'}
set()
Traceback (most recent call last):

```
  File "<string>", line 31, in <module>
NameError: name 'Language' is not defined
```

II.

```
{'Chasis': 'Hatchback', 'Engine': 'VCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol'}
VCylinder
VCylinder
{'Chasis': 'Hatchback', 'Engine': 'VCylinder', 'Capacity': '1000cc', 'Variant': 'Petrol',
'Cost': '75000'}
{'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc', 'Variant':
'Petrol', 'Cost': '75000'}
75000
{'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc', 'Variant':
'Petrol'}
('Variant', 'Petrol')
{'Chasis': 'Hatchback', 'Engine': 'TwinCylinder', 'Capacity': '1000cc'}
{'Chasis': 'Hatchback', 'Capacity': '1000cc'}
{}
Traceback (most recent call last):
  File "<string>", line 35, in <module>
NameError: name 'automobile' is not defined
```

III.

```
{'Pillars': '100', 'Floors': '5', 'Windows': '50'}
Contents of the Dictionary = {'Pillars': '100', 'Floors': '5', 'Windows': '50'}
Checking if Pillars is a key in dictionary = True
Checking if Walls is not a key in dictionary = True
Cannot check with pair value 50 in dictionary (True - pair can be checked, False -
pair cannot be checked = False
Pair values in the dictionary =
100
5
50
Length of the dictionary = 3
Sort the keys of the dictionary in alphabetical order = ['Floors', 'Pillars', 'Windows']
```

9. Result

The operations for the sets and dictionaries were successfully implemented through python program.

10. Viva Questions

A. How to access the item in a set?

We cannot access items in a set by referring to an index, since sets are unordered the items has no index. But We can loop through the set items using a for loop, or ask if a specified value is present in a set, by using the in keyword.

B. What will be the output, if you try to remove an item which is not in the list ?

If the item to remove does not exist, remove() will raise an error.
If the item to remove does not exist, discard() will NOT raise an error.

C. What is the difference between del and clear() in sets?

The clear() method empties the set, whereas del keyword will delete the set completely:

D. How to add or change elements in a dictionary?

Dictionary is mutable. We can add new items or change the value of existing items using assignment operator. If the key is already present, value gets updated, else a new key: value pair is added to the dictionary.

E. What is the difference between pop(key[,d]) and popitem() in dictionary?

pop(key[,d]) - Remove the item with key and return its value or d if key is not found. If d is not provided and key is not found, raises KeyError.
popitem() - Remove and return an arbitrary item (key, value). Raises KeyError if the dictionary is empty.

F. What is membership test for the elements in dictionary?

We can test if a key is in a dictionary or not using the keyword in. The membership test is for keys only, not for values. For example,

```
squares = {1: 1, 3: 9, 5: 25, 7: 49, 9: 81}
# Output: True
print(1 in squares)
# Output: True
print(2 not in squares)
# membership tests for key only not value
# Output: False
print(49 in squares)
```

Ex.No: 6 Implementing Programs using Functions

Date :

1. **Problem Statement**: Write a python program to implement Factorial of a number, largest number in a list, area of shape using functions.

2. **Expected Learning Outcomes**: Train the students to understand the basics of functions.

3. **Problem Analysis**:

    a. Using function concepts implement Factorial of a non-negative integer, is multiplication of all integers smaller than or equal to n.
    b. Get the number of elements from user. Using function identify the largest number in the given list.
    c. Using Python create a module that contains area calculation functions of different shapes (triangle, circle and rectangle)

4. **A. Expected Input:**

I. Enter the value of N: 3

II. Enter the list of elements: 7, 2, 23, 3, 19

III.Enter name of the shape: Rectangle
Enter rectangle's length: 4
Enter rectangle's breadth:3
Enter name of the shape: Triangle
Enter triangle's height length: 4
Enter triangle's breadth length: 3
Enter name of the shape: Square
Enter square's side length: 4
Enter name of the shape: Circle
Enter circle's radius length: 4

B. Expected Output:

I. The factorial of the given number 3 is: 6

II. The largest number in the list is: 23

III. Area of Rectangle:12
Area of Triangle: 6
Area of Square: 16
Area of Circle: 50.24

5. **Algorithm:**

I.

Step 1: Start
Step 2: Declare Variable n, fact, i
Step 3: Read number from User
Step 4: Initialize Variable fact=1 and i=1
Step 5: Repeat Until i<=number
    5.1 fact=fact*i
    5.2 i=i+1

Step 6: Print fact
Step 7: Stop

II.

Step 1: Start
Step 2: Create an empty list
Step 3: Read list elements from User
Step 4: Assign first number in list to variable "max"
Step 5: Use for loop compare each number with "max" value
Step 6: Assign the largest value to "max'
Step 7: Repeat Step 5 and 6 until reach the last element of the list
Step 8: Print max as largest number in the list

III.

Step 1: Select the shape
Step 2: If it is rectangle do the following
    Step 2.1: Enter the width of the rectangle.
    Step 2.2: Enter the Height of the rectangle.
    Step 2.3: Calculate the area of the rectangle by multiplying the width and height of the rectangle.
    Step 2.4: Assign the area of the rectangle to the area variable.
    Step 2.5: print the area of the rectangle.
Step 3: If it is triangle do the following
    Step 3.1: Enter the breadth of the triangle
    Step 3.2: Enter the Height of the triangle
    Step 3.3: Calculate the area of the rectangle by multiplying the breadth and height of the triangle
    Step 3.4: Assign the area of the triangle to the area variable.
    Step 3.5: print the area of the triangle
Step 4: If it is square do the following
    Step 4.1: Enter the side of the square
    Step 4.2: Calculate the area of the square by multiplying the 4 sides
    Step 4.3: Assign the area of the Square to the area variable.
    Step 4.4: Print the area of the square
Step 5: If it is circle do the following

Step 5.1: Enter the radius of the circle
Step 5.2: Calculate the area of the circle by $A = \pi r^2$
Step 5.3: Assign the area of the circle to the area variable.
Step 5.4: Print the area of the circle

6. **Coding:**

a. **Write a program to find the factorial of a number.**

```
# factorial of given number
def factorial(n):
    if n < 0:
        return 0
    elif n == 0 or n == 1:
        return 1
    else:
        fact = 1
        while(n > 1):
            fact *= n
            n -= 1
        return fact
 num = 5;
print("Factorial of",num,"is",
factorial(num))
```

b. **Write a program to find the largest number in the list.**

```
 # Python program to find largest number in a list

# creating empty list
list1 = []

# asking number of elements to put in list
num = int(input("Enter number of elements in list: "))

# iterating till num to append elements in list
for i in range(1, num + 1):
    ele = int(input("Enter elements: "))
    list1.append(ele)

# print maximum element
print("Largest element is:", max(list1))
```

C. Write a program to find area of different shapes.

```
 # define a function for calculating  the area of a shapes
 def calculate_area(name):\

  # converting all characters  into lower cases
```

```
  name = name.lower()

  # check for the conditions
  if name == "rectangle":
    l = int(input("Enter rectangle's length: "))
    b = int(input("Enter rectangle's breadth: "))

    # calculate area of rectangle
    rect_area = l * b
    print(f"Area of rectangle is
        {rect_area}.")

  elif name == "square":
    s = int(input("Enter square's side length: "))

    # calculate area of square
    sqt_area = s * s
    print(f"Area of square is
        {sqt_area}.")

  elif name == "triangle":
    h = int(input("Enter triangle's height length: "))
    b = int(input("Enter triangle's breadth length: "))

    # calculate area of triangle
    tri_area = 0.5 * b * h
    print(f"Area of triangle is
        {tri_area}.")

  elif name == "circle":
    r = int(input("Enter circle's radius length: "))
    pi = 3.14

    # calculate area of circle
    circ_area = pi * r * r
    print(f"The area of triangle is
        {circ_area}.")
        else:
    print("Sorry! This shape is not available")

# driver code
if __name__ == "__main__" :

  print("Calculate Shape Area")
shape_name = input("Enter the name of shape whose area you want to find: ")
```

```
# function calling
calculate_area(shape_name)
```

7. Test case :

	Input	**Output**
I	Enter the value of N:4	The factorial of the given number 4 is: 24
	Enter the value of N:-3	The factorial of the given number -3 is:
	Please enter the value of N: a	invalid literal for int() with base 10: 'a'
II	Enter the list of elements:[5,8,3 9,2]	The largest number in the list is 9
	Enter the list of elements:[33,25,28,34 19,2]	The largest number in the list is 34
III	Enter name of the shape: Rectangle	
	Enter rectangle's length: 4	
	Enter rectangle's breadth:3	Area of Rectangle:12
	Enter name of the shape: Triangle	
	Enter triangle's height length: 4	
	Enter triangle's breadth length: 3	Area of Triangle: 6
	Enter name of the shape: Square	
	Enter square's side length: 4	Area of Square: 16
	Enter name of the shape: Circle	
	Enter circle's radius length: 4	Area of Circle: 50.24
	Enter rectangle's length: s	could not convert string to float: 's'

8. Output :

I. The factorial of the given number 3 is: 6

II. The largest number in the list is: 23

III. Area of Rectangle:12
Area of Triangle: 6
Area of Square: 16
Area of Circle: 50.24

9. Result

Factorial of a number, Largest number in a list, Area of different shapes are successfully implemented through functions.

10. Viva Questions

A. Define range() function.

The range() Function

To loop through a set of code a specified number of times, we can use the range() function,

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

B. Define function.

A function is a block of code which only runs when it is called.You can pass data, known as parameters, into a function.A function can return data as a result.

C. How to create a Function in python?

In Python a function is defined using the def keyword:

D. How to Call a Function in python?

To call a function, use the function name followed by parenthesis:

E. What is Recursion?

Python also accepts function recursion, which means a defined function can call itself.

Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

F. Why Use Lambda Functions?

The power of lambda is better shown when you use them as an anonymous function inside another function.Say you have a function definition that takes one argument, and that argument will be multiplied with an unknown number.

Ex.No: 7 Implementing programs using Strings

Date :

1. **Problem Statement**: Write a python program to Implementing programs (reverse, palindrome, character count, replacing characters using Strings

2. **Expected Learning Outcomes**: Train the students to understand the basics of Strings concepts.

3. **Problem Analysis**:

    c. Using string concepts to implement reverse a string.
    d. Using string concepts to implement to check the given string is palindrome or not.
    e. Using string concepts to count the number of characters in a given string.
    f. Using string concepts to replace characters in a given string

4. **A. Expected Input:**

    I. Please enter the string a:welcome

    II. Please enter the string x: madam

    III. Please enter the string x:knowledge
    Please enter the char to be count :e

    IV. Please enter the string x:good
    Please enter which the char to be replace :o a

    Please enter how many times u want to replace the char:1

    B. Expected Output:

    I. Reverse string is: emoclew

    II. Yes,The string is Palindrome

    III. count of e in given string is: 2

    IV. the replaced string is gaad

5. **Algorithm:**

I.

Step 1: Get the input from the user and store the value in a
Step 2: Use slice statement and read the string
. Step 3: Print the reverse String

II.

Step 1: Get the input string from user and store in x
Step 2: Define a palindrome function
Step 3: Use slicing function and reverse the string
Step 4: check both the string are same
Step 5: If yes, The string is Palindrome
Step 6: If not, The string is not a Palindrome

III.

Step 1: Read the string from user and store in x
Step 2: Read the char to be count and store in y
Step 3: Use count function to count the number of characters
Step 4: Display the output

IV.

Step 1: Read the string from user and store in x
Step 2: Read the char to be replaced and store in a,b
Step 3: Use replace function to replace the characters
Step 4: Display the output

6. **Coding:**

I. Write a program to print Reverse String

```
a=input("please enter the string a:")
x=a[::-1]
print("Reverse string is:",x)
```

II. Write a program to display given string is palindrome or not

```
def isPalindrome(x):
        return x == x[::-1]
x = input("Please enter the string x:")
y = isPalindrome(x)
if y:
        print("Yes,The string is Palindrome ")
else:
        print("No,The string is  not a Palindrome")
```

III. Write a program to count a character from a given string

```
x = input("Please enter the string x:")
y = input("Please enter the char to be count :")
counter = x.count(y)
print("count of " + y +" in given string is:", str(counter))
```

IV. Write a program to replace a character from a given string

```
x = input("Please enter the string x:")
a,b = input("Please enter which the char to be replace :").split()
re = x.replace(a,b)
print("the replaced string is ",re)
```

7. Test case :

	Input	Output
I	Please enter the string a:welcome	Reverse string is: emoclew
	Please enter the string a: python programming	Reverse string is: gnimmargorp nohtyp
II	Please enter the string x:malayalam	Yes,The string is Palindrome
	Please enter the string x:modem	No,The string is not a Palindrome
III	Please enter the string x:knowledge Please enter the char to be count :e	count of e in given string is: 2
	Please enter the string x:python Please enter the char to be count :g	count of g in given string is: 0
IV	Please enter the string x:good Please enter which the char to be replace :o a Please enter how many times u want to replace the char:1	the replaced string is gaad the replaced string is gaod

8. Output :

I. Reverse string is: gnimmargorp nohtyp

II. Yes,The string is Palindrome

III. count of e in given string is: 2

IV. the replaced string is gaad

9. Result

The programs Reverse a string , palindrome, character count, replacing characters using Strings are successfully implemented.

10. Viva Questions

A. What is docstring in Python?

Documentation string or docstring is a multiline string used to document a specific code segment.

The docstring should describe what the function or method does

B. What is slicing in Python?

- As the name suggests, ‘slicing’ is taking parts of.
- Syntax for slicing is **[start : stop : step]**
- **start** is the starting index from where to slice a list or tuple
- **stop** is the ending index or where to sop.
- **step** is the number of steps to jump.
- Default value for **start** is 0, **stop** is number of items, **step** is 1.
- Slicing can be done on **strings, arrays, lists**, and **tuples**.

C. What is the difference between Python Arrays and lists?

- Arrays in python can only contain elements of same data types i.e., data type of array should be homogeneous. It is a thin wrapper around C language arrays and consumes far less memory than lists.
- Lists in python can contain elements of different data types i.e., data type of lists can be heterogeneous. It has the disadvantage of consuming large memory.

D. What are decorators in Python?

Decorators in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function itself. They are represented the @decorator_name in Python and are called in a bottom-up fashion.

E. What are Dict and List comprehensions?

Python comprehensions, like decorators, are **syntactic sugar** constructs that help **build altered** and **filtered lists**, dictionaries, or sets from a given list, dictionary, or set. Using comprehensions saves a lot of time and code that might be considerably more verbose (containing more lines of code).

F. What is lambda in Python? Why is it used?

Lambda is an anonymous function in Python, that can accept any number of arguments, but can only have a single expression. It is generally used in situations requiring an anonymous function for a short time period.

Ex.No: 8 Programs using written modules and Python Standard Libraries

Date :

5. **Problem Statement:** Write a python program using python runnable code called modules and use python standard libraries.

6. **Expected Learning Outcomes**: Train the students to understand the basic of modules and libraries available in python

7. **Problem Analysis:**
   I. Use python math module to compute various mathematical operations.
   II. Compute the Mean, Median, Mode, Quartiles and Standard deviation using Numpy and pandas libraries.

8. **A. Expected Input:**

   I. Enter the set of 3 numbers for finding the square root, Factorial and log base2 :3,5,10
   II. Take a set of eight numbers as input.
   Set of frequencies: [11, 21, 34, 22, 27, 11, 23, 21]

B. Expected Output:

IV. Square root 1.7320508075688772
Factorial 3628800

Log base 2 2.321928094887362

V.

Mean= 21.25
Mean= 21.25
Median 21.5
Mode [11, 21]
Mode 11
Quartiles (16.0, 21.5, 25.0)
Standard Deviation Simple Method: 7.1545440106270926
Standard Deviation using Numpy: 7.1545440106270926

5. Algorithm:

IV. Step 1: Get three Numbers from user and store it in a, b, and c
Step 2: Compute Square root of a and display the result

Step 3: Compute factorial of b and display the result
Step 4: Compute log base 2 of c and display the result
Step 5: Stop

V.

Step1: Take a list of 8 Numbers.
Step2: Compute the Mean value by simple Computation and print it.
Step3: Compute the Mean value using numpy method and print it.
Step4: Compute the Median value by simple Computation and print it.
Step5: Compute the Mode value by simple Computation and print it.
Step6: Compute the Mode value using numpy method and print it.
Step7: Compute the IQR (Inter Quartile Range) by simple Computation and print it.
Step8: Compute the Standard Deviation by simple Computation and print it.
Step9: Compute the Standard Deviation using Numpy and print it.
Step 10: Stop

6. Coding:

IV. Write a program to compute Square root , Factorial and Log base 2 using math module in python

```
import math as
a,b,c = input('Enter the set of 3 numbers for finding the square root , Factorial and log base2 :').split(',')
a = int(a)
b = int(b)
c = int(c)
print('Square root ',m.sqrt(a))
print('Factorial ',m.factorial(b))
print('Log base 2 ',m.log2(c))
```

V. Write a program to compute mean, median, mode, IQR and Standard Deviation using Numpy, Pandas and Scipy in python

```
import numpy as n
from collections import Counter
from scipy import stats
# Finding Mean by simple Computation
a= [11, 21, 34, 22, 27, 11, 23, 21]
mean = sum(a)/len(a)
print (mean)
# Finding Mean using numpy method
mean = np.mean(a)
print (mean)
# Finding Median by simple Computation.
```

```
def median(nums):
nums.sort()
if len(nums)%2 == 0:
return int(nums[len(nums)//2-1]+nums[len(nums)//2])/2
else:
return nums[len(nums)//2]
print (median(a))
# Finding Mode by simple Computation
data = dict(Counter(a))
mode = [k for k, v in data.items() if v == max(list(data.values()))]
print (mode)
# Finding Mode using numpy method
print (stats.mode(a)[0][0])
# Finding Quartiles by simple method
def quartiles(nums):
nums=sorted(nums)
Q1 = median(nums[:len(nums)//2])
Q2 = median(nums)
if len(nums)%2 == 0:
Q3 = median(nums[len(nums)//2:])
else:
Q3 = median(nums[len(nums)//2+1:])
return Q1,Q2,Q3
def median(nums):
nums.sort()
if len(nums)%2 == 0:
return int(nums[len(nums)//2-1]+nums[len(nums)//2])/2
else:
return nums[len(nums)//2]
print (quartiles(a))
# Find Standard deviation by simple computation
n=len(a)
std=(sum(map(lambda x: (x-sum(a)/n)**2,a))/n )**0.5
print(std)
# Find Standard deviation using numpy method
print (np.std(a))
```

8. Test case:

	Input	Output
II.	0,0,0	Square root 0.0 Factorial 1 ValueError: math domain error
	100,2,100	Square root 10.0 Factorial 2 Log base 2 6.643856189774724

III.	[9, 8, 31, 20, 0, 31, 31, 20]	Mean = 18.75 Mean = 18.75 Median = 20.0 Mode = [31] Mode = 31 Quartiles = (8.5, 20.0, 31.0) Strandard Deviation = 11.266654339243749 Strandard Deviation = 11.266654339243749
	['a', 100, 310, 200, 100, 1, 100, 20]	TypeError: cannot perform reduce with flexible type

11. Output

I. Square root 1.7320508075688772
Factorial 3628800

Log base 2 2.321928094887362

II. Mean= 21.25
Mean= 21.25
Median 21.5
Mode [11, 21]
Mode 11
Quartiles (16.0, 21.5, 25.0)
Standard Deviation Simple Method: 7.1545440106270926
Standard Deviation using Numpy: 7.1545440106270926

12. Result

Few most important modules including Math, Numpy array, pandas and Scipy were successfully implemented through python program.

13. Viva Questions

A. What is a module in python?

Modules refer to a file containing Python statements and definitions.
A file containing Python code, for example: example.py, is called a module, and its module name would be example.
We use modules to break down large programs into small manageable and organized files. Furthermore, modules provide reusability of code.
We can define our most used functions in a module and import it, instead of copying their definitions into different programs.

B. What are the different ways to import a module in python?

a. import module.
b. from module import function.
c. from module import *

C. What is import * in Python?

Use the asterisk (*) as a wild card. The following statement will import every function and property contained in the math package.

from module import *

D. What is the use of NumPy in Python?

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

E. Why pandas is used in Python?

Pandas is a Python library for data analysis. ... Pandas is built on top of two core Python libraries—matplotlib for data visualization and NumPy for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code.

F. What is SciPy in Python used for?

SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.

Ex.No: 9 Implementing real-time/technical applications using File handling.

Date :

1. **Problem Statement**: Write a python program to implement real-time/technical applications using File handling. (copy from one file to another, word count, longest word)

2. **Expected Learning Outcomes**: Train the students to understand the basics of file handling operations.

3. **Problem Analysis**:

    a. Using read, write & append operations in file concepts implement the copy contents of the first file into the second file
    b. Using split() function implement the word count from a given text or sentence.
    c. Using string split() function & max() function implement the longest word from sentence

4. **A. Expected Input:**

    a. Myinput file.txt: Hi this is first copy file python program
    Myoutput file.txt :
    b. The original string is : Tutorials point is a learning platform.
    c. Enter sentence: Tongue tied and twisted just an earth bound misfit I

    B. Expected Output:

    I. Myoutput file.txt: Hi this is first copy file python program
    II. The number of words in string are : 6
    III. Longest word is: twisted
    And its length is: 7

5. **Algorithm:**

    I.
    Step1: Start
    Step2: Create text files first.txt and save with some contents & second.txt with Empty contents
    Step3: open first.txt in 'r' mode
    Step2: read the contents of first.txt.
    Step3: open second.txt in 'a' mode and will append the content of first.txt into second.txt.
    Step 6: open second.txt
    Step 7: Stop

    II.
    Step 1: Start
    Step 2: Get the string from User

Step 3: Read list elements from User
Step 4: Use Split function breaks the string into a list iterable with space as a delimiter
Step 5: if the split() function is used without specifying the delimiter space is allocated as a default delimiter.
Step 6: Print number of words in string

III.

Step 1: Start
Step 2: Get the string from User
Step 3: Read sentence from user
Step 4: Use Split function to convert it to list.
Step 5: After splitting to use max() function with keyword argument key=len
Step 6: Print longest word from sentence

6. Coding:

a. **Write a program to copy from one file to another.**

```
# Creating an output file in writing mode
output_file = open("myoutput file.txt", "w")
 # Opening input file and scanning each line
# from input file and writing in output file
with open("myinput file.txt", "r") as scan:
    output_file.write(scan.read())
 # Closing the output file
output_file.close()
```

b. **Write a program to find find word count from a given text or sentence.**

```
test_string = "Tutorials point is a learning platform"
#original string
print ("The original string is : " + test_string)
# using split() function
res = len(test_string.split())
# total no of words
print ("The number of words in string are : " + str(res))
```

c. **Write a program to find longest word from sentence**

```
 # Longest word
# Reading sentence from user
sentence = input("Enter sentence: ")
# Finding longest word
longest = max(sentence.split(), key=len)
# Displaying longest word
print("Longest word is: ", longest)
print("And its length is: ", len(longest))
```

7. Test case :

	Input	**Output**
i	Myinputfile.txt: Hi this is my first copy file python program	Myoutput file.txt: Hi this is my first copy file python program
	Myinputfile.txt: 7764gffghhg	Myoutputfile.txt: 7764gffghhg
ii	The original string is : Tutorials point is a learning platform	The number of words in string are : 6
	The original string is : This is my test string for word count problem	The number of words in string are : 8
iii	Enter sentence: Tongue tied and twisted just an earth bound misfit I	Longest word is: twisted And its length is: 7
	Enter sentence: This is my test string for longest word count exercise	Longest word is: exercise And its length is: 8

8. Output :

i. Myoutput file.txt: Hi this is first copy file python program

ii. The number of words in string are : 6

iii. Longest word is: twisted
And its length is: 7

9. Result

Real-time/technical applications (copy from one file to another, word count, longest word) implemented through File handling operations.

10. Viva Questions

G. Write short hints about access modes in files?.

Access modes govern the type of operations possible in the opened file. It refers to how the file will be used once its opened. These modes also define the location of the File Handle in the file. File handle is like a cursor, which defines from where the data has to be read or written in the file.

H. List out some of the access modes are used in files .

- Read Only (‘r’)
- Write Only (‘w’)
- Read and Write (‘r+’)
- Write and Read (‘w+’)
- Append Only (‘a’)

- Append and Read (‘a+’)

I. How to use opening a file in python?

It is done using the open() function. No module is required to be imported for this function.

File_object = open(r"File_Name","Access_Mode")

The file should exist in the same directory as the python program file else, full address of the file should be written on place of filename.

J. How to use closing a file in python?

close() function closes the file and frees the memory space acquired by that file. It is used at the time when the file is no longer needed or if it is to be opened in a different file mode.

File_object.close()

Ex.No : 10 Real-time/technical applications using Exception handling

Date :

1. **Problem Statement**: Write a python program to Implement programs real-time/technical applications using Exception handling. (divide by zero error, voter's age validity, student mark range validation)
2. **Expected Learning Outcomes**: Train the students to understand the basics of Exception handling

3. **Problem Analysis**:

    a. Using division concepts to implement ZeroDivisionError.
    b. Using conditional check concepts to implement voter's age validity
    c. Using Average computation concepts to find a student mark range validation

4. **A. Expected Input:**

    a. Enter a:10
    Enter b:0
    b. Enter Age : 19
    c. Enter marks of the first subject: 85
    Enter marks of the second subject: 95
    Enter marks of the third subject: 99
    Enter marks of the fourth subject: 93
    Enter marks of the fifth subject: 100

    B. Expected Output:

    I. ZeroDivisionError: division by zero
    II. You are Eligible for Vote
    III. Grade: A

5. **Algorithm:**

    a.
    Step 1: Get the input from the user and store the value in a & b
    Step 2: Use try & catch with division statement and read the input values
    . Step 3: Print the ZeroDivisionError

    b.
    Step 1: Get the input from user and store in x
    Step 2: Use a conditional check statement and read the string

Step 3: If yes, You are Eligible for Vote
Step 6: If not, You are Not Eligible for Vote

III.

Step 1: Get the input from the user and store the value in sub1, sub2 ,sub3, sub4 & sub5
Step 3: Use Avg computation and to get Avg value
Step 4: Display the output

6. Coding:

V. Write a program to print ZeroDivisionError

```
try:
    a = int(input("Enter a:"))
    b = int(input("Enter b:"))
    c = a/b
    print("a/b = %d"%c)
    # Using exception object with the except statement
except Exception as e:
    print("can't divide by zero")
    print(e)
else:
    print("Hi I am else block")
```

VI. Write a program to display voter's age validity

```
 # input age
age = int(input("Enter Age : "))

# condition to check voting eligibility
if age>=18:
       status="Eligible"
else:
    status="Not Eligible"

print("You are ",status," for Vote.")
```

VII. Write a program to find a student mark range validation

```
sub1=int(input("Enter marks of the first subject: "))
sub2=int(input("Enter marks of the second subject: "))
sub3=int(input("Enter marks of the third subject: "))
sub4=int(input("Enter marks of the fourth subject: "))
sub5=int(input("Enter marks of the fifth subject: "))
```

```
avg=(sub1+sub2+sub3+sub4+sub4)/5
if(avg>=90):
   print("Grade: A")
elif(avg>=80&avg<90):
   print("Grade: B")
elif(avg>=70&avg<80):
   print("Grade: C")
elif(avg>=60&avg<70):
   print("Grade: D")
else:
   print("Grade: F")
```

7. Test case :

	Input	**Output**
I	Enter a:10 Enter b:0	ZeroDivisionError: division by zero
	Enter a:10 Enter b:5	Will not throw division by zero
I	Enter Age : 19	Yes,You are Eligible for Vote
	Enter Age : 17	No, You are not Eligible for Vote
III	Enter marks of the first subject: 85 Enter marks of the second subject: 95 Enter marks of the third subject: 99 Enter marks of the fourth subject: 93 Enter marks of the fifth subject: 100	Grade: A
	Enter marks of the first subject: 81 Enter marks of the second subject: 72 Enter marks of the third subject: 94 Enter marks of the fourth subject: 85 Enter marks of the fifth subject: 80	Grade: B

8. Output :

I. ZeroDivisionError: division by zero
II. You are Eligible for Vote
III. Grade: A

9. Result

Thus The programs real-time/technical applications using Exception handling(divide by zero error, voter's age validity, student mark range validation) are successfully implemented.

10. Viva Questions

G. What is Exception Handling?

An exception can be defined as an unusual condition in a program resulting in the interruption in the flow of the program.

Whenever an exception occurs, the program stops the execution, and thus the further code is not executed.

H. List out some common exceptions?

A list of common exceptions that can be thrown from a standard Python program is given below.

- ZeroDivisionError: Occurs when a number is divided by zero.
- NameError: It occurs when a name is not found. It may be local or global.
- IndentationError: If incorrect indentation is given.
- IOError: It occurs when Input Output operation fails.
- EOFError: It occurs when the end of the file is reached, and yet operations are being performed..

I. What is the difference between Exception and without the Exception?

- We can use the exception variable with the except statement. It is used by using the as keyword. this object will return the cause of the exception.
.
- Without the Exception, It will not interrupt the program execution and will not through a exception message.

Ex.No: 11 Exploring Pygame

Date :

1. **Problem Statement**: Exploring a cross-platform that is used to develop video games.

2. **Expected Learning Outcomes**: As a beginner, it helps students to understand the basic concepts of Pygame and how to install and use this module.

3. Description

Pygame

- Pygame is a cross-platform set of Python modules which is used to create video games.
- It consists of computer graphics and sound libraries designed to be used with the Python programming language.
- Pygame was officially written by **Pete Shinners** to replace PySDL.
- Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

Prerequisites for Pygame:

Before learning about pygame, we need to understand what kind of game we want to develop.

- To learn pygame, it is required to have basic knowledge of Python.

Pygame Installation

Install pygame in Windows

Before installing Pygame, Python should be installed in the system, and it is good to have 3.6.1 or above version because it is much friendlier to beginners, and additionally runs faster. There are mainly two ways to install Pygame, which are given below:

1. **Installing through pip:** The good way to install Pygame is with the pip tool (which is what python uses to install packages). The command is the following:
PIP is a tool that is used to install python packages. PIP is automatically installed with Python 2.7. 9+ and Python 3.4+. Open the command prompt and enter the command shown below to check whether pip is installed or not.

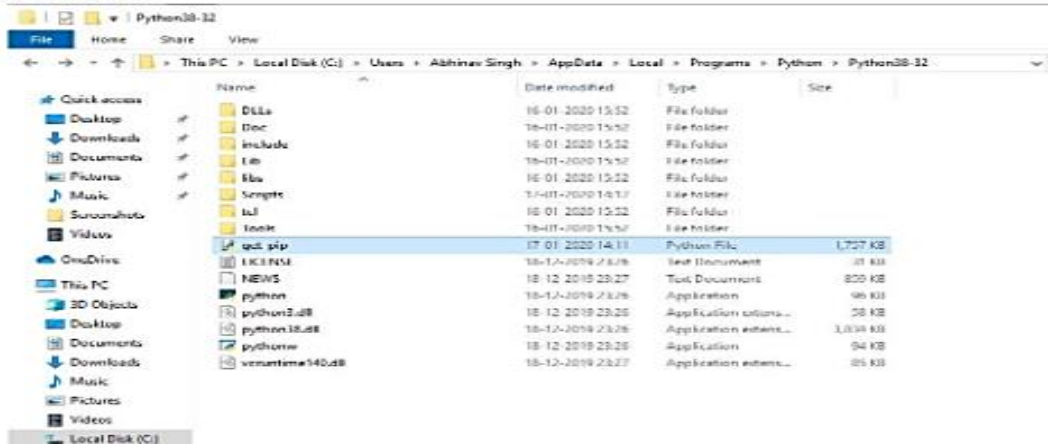
pip install package_name

pip will look for that package on PyPI and if found, it will download and install the package on your local system.

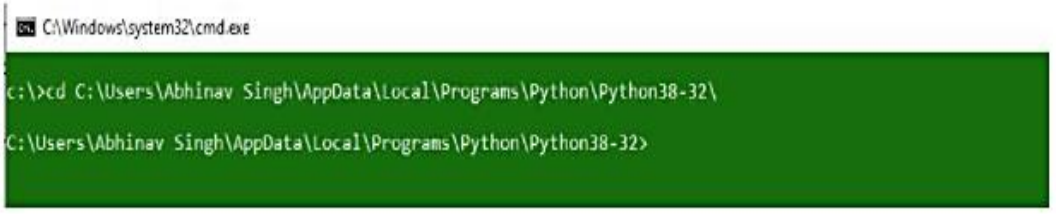
Download and Install pip:

pip can be downloaded and installed using command-line by going through the following steps:

- Download the get-pip.py file and store it in the same directory as python is installed.



Change the current path of the directory in the command line to the path of the directory where the above file exists.



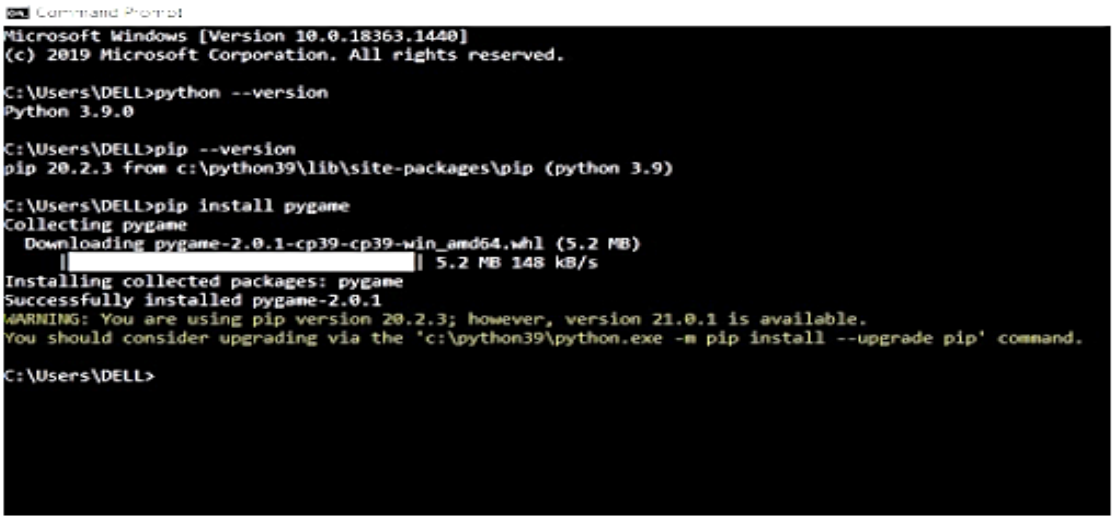
Run the command given below:
python get-pip.py

and wait through the installation process.

pip is now installed on your system.

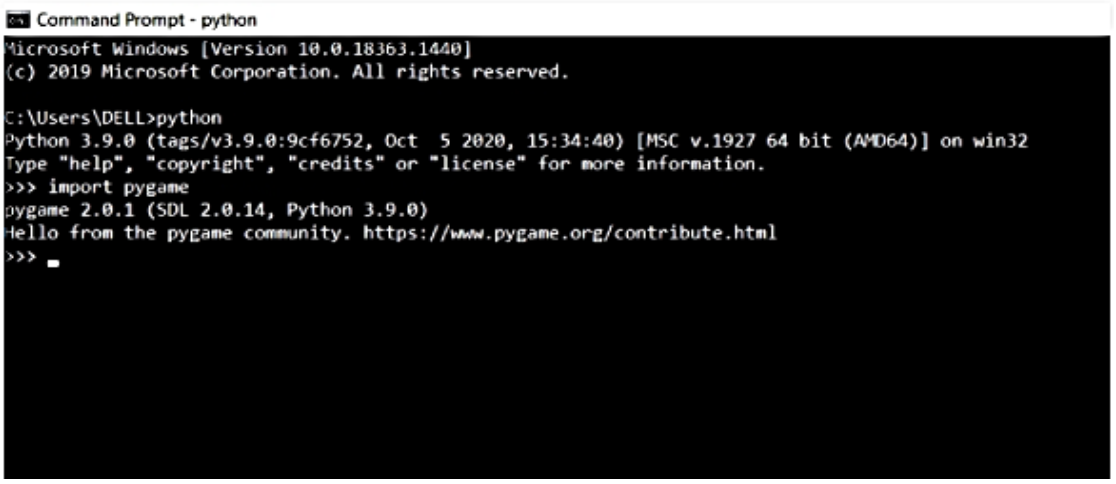
Step 3: Install Pygame
To install Pygame, open the command prompt and give the command as shown below:

pip install pygame



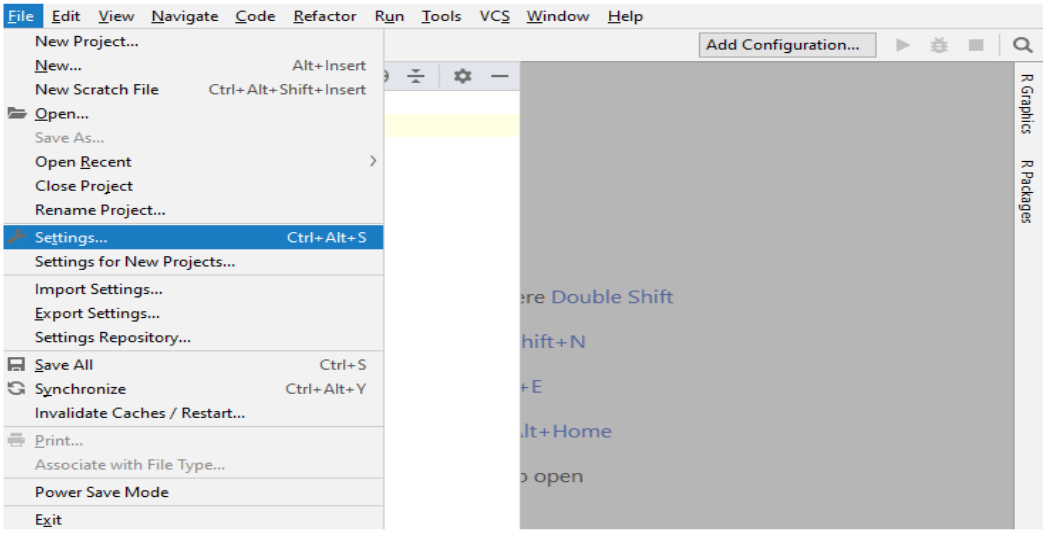
Step 4: Check Whether PyGame is Working or not

Now open a new terminal and import the Pygame library to see whether it is working fine or not in our system. The library is imported successfully means we got success.

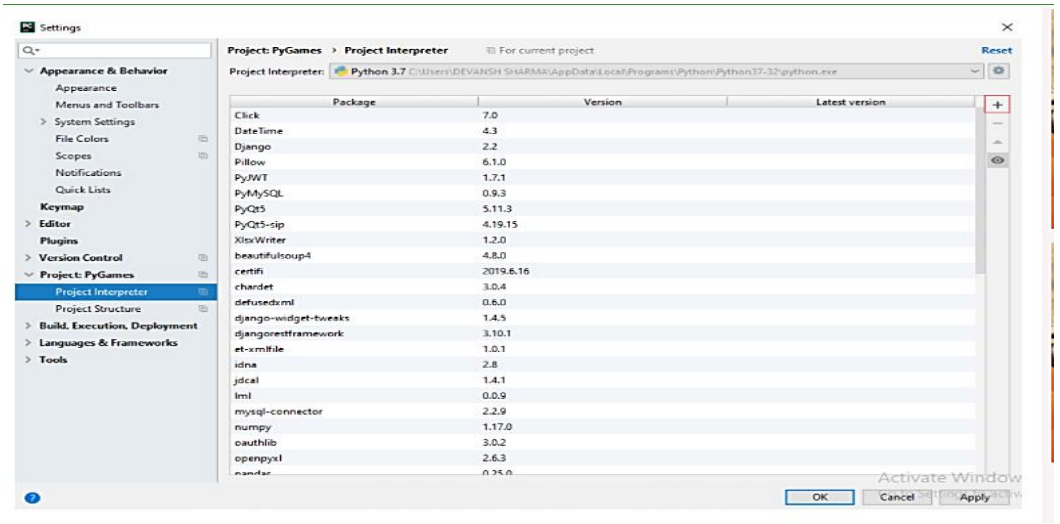


Installing through an IDE: The second way is to install it is through an IDE and here we are using Pycharm IDE. Installation of pygame in the pycharm is straightforward. We can install it by running the above command in the terminal or use the following steps:

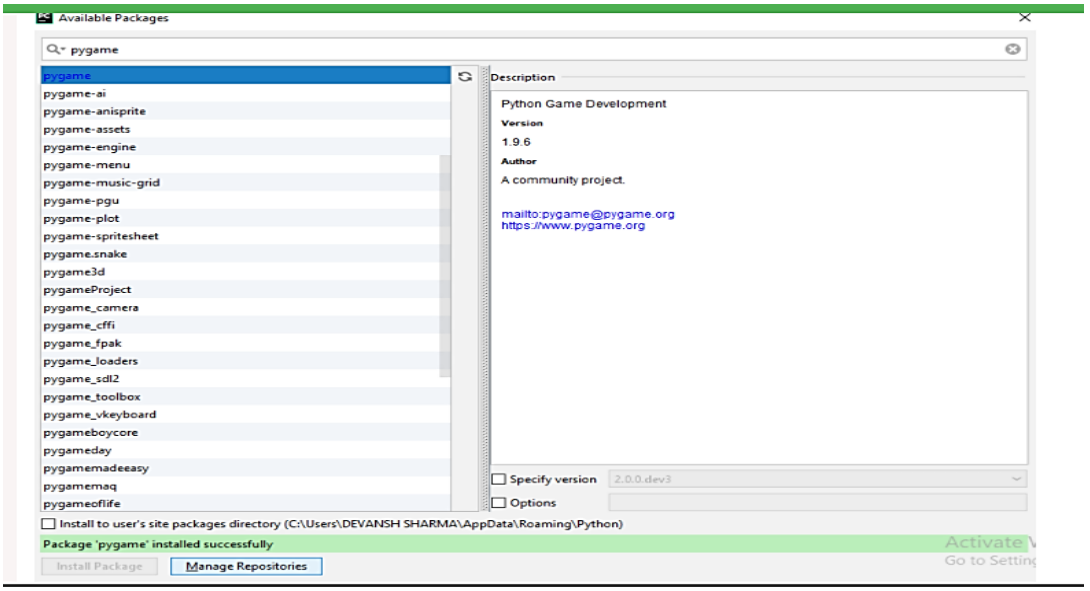
- Open the **File** tab and click on the **Settings** option.



- Select the **Project Interpreter** and click on the **+** icon.



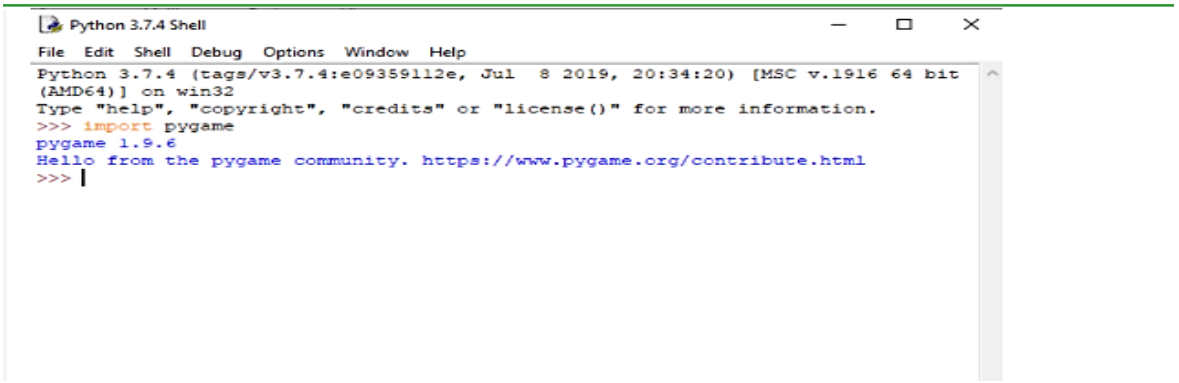
- It will display the search box. Search the pygame and click on the **install package** button.



To check whether the pygame is properly installed or not, in the IDLE interpreter, type the following command and press Enter:

import pygame

If the command runs successfully without throwing any errors, it means we have successfully installed Pygame and found the correct version of IDLE to use for pygame programming.



Basic pygame methods

pygame.init() - This is used to initialize all the required module of the pygame.

pygame.display.set_mode((width, height)) - This is used to display a window of the desired size. The return value is a Surface object which is the object where we will perform graphical operations.

pygame.event.get()- This is used to empty the event queue. If we do not call this, the window messages will start to pile up and, the game will become unresponsive in the opinion of the operating system.

pygame.QUIT - This is used to terminate the event when we click on the close button at the corner of the window.

pygame.display.flip() - Pygame is double-buffered, so this shifts the buffers. It is essential to call this function in order to make any updates that you make on the game screen to make visible.

Pygame Blit

The pygame blit is the process to render the game object onto the surface, and this process is called **blitting**. When we create the game object, we need to render it. If we don't render the game objects and run the program, then it will give the black window as an output.

Blitting is one of the slowest operations in any game so, we need to be careful to not to blit much onto the screen in every frame. The primary function used in blitting is blit(), which is:

blit()

Pygame Adding Image

To add an image on the window, first, we need to instantiate a blank surface by calling the Surface constructor with a width and height tuple.

surface = pygame.Surface((100,100))

The above line creates a blank 24-bit RGB image that's 100*100 pixels with the default black color.

For the transparent initialization of Surface, pass the SRCALPHA argument.

surface = pygame.Surface((100,100), pygame.SRCALPHA)

4. Result

Thus to study and explore Pygame was successfully completed.

5. Viva Questions

A. Can PyGame make 3D games?

No, Pygame is a wrapper for SDL, which is a 2D api. Pygame doesn't provide any 3D capability and probably never will.

B. Is Pygame a game engine?

It is an advanced, feature-packed, multi-platform 2D and 3D open source game engine.

C. How do you make your character jump in Pygame?

Start a jump by pressing the space bar (set a variable) and stop it, when you touch the floor. you start a jump on keydown (pygame. KEYDOWN) not if pressed.

D. How do you Blit an image in pygame?

Create a Image surface object i.e.surface object in which image is drawn on it, using image. **load() method** of pygame. Copy the image surface object to the display surface object using blit() method of pygame display surface object. Show the display surface object on the pygame window using display.

E. How do you flip an image in pygame?

To flip the image we need to use pygame. transform. flip(Surface, xbool, ybool) method which is called to flip the image in vertical direction or horizontal direction according to our needs

Ex.No: 12 Developing a game activity using Pygame

Date :

1. **Problem Statement**: Write a python program to develop a game activity like bouncing ball, elliptical orbit etc. using Pygame

2. **Expected Learning Outcomes**: As a beginner, it helps students to create dynamic scripts and to build their first game using Pygame.
   - Once learning coding games, it gains insights into the complex processes that go into building games.
   - It ensures the students to create a code with better readability with lesser code lines and better design

3. **Problem Analysis**:

   a. Using **pygame. image. Load()** the ball image has been opened and an infinite loop has been created to make the ball move continuously and reverse the direction of ball if it hits the edges. Pygame provides **move()** method to make an object move smoothly across the screen.Using **fill()** method the surface background color is filled.we also need to render the moving object on the window In pygame, this is called blitting of an object and implemented with **blit()** method. **flip ()** method is used to make all image visible.

   b. Using display.setmode() set the screen size .Let us set the x and y radius of ellipse. starting from degree 0 ending with 360 degrees in increments of 10 degrees calculate the (x1, y1) coordinates to find a point in the elliptical orbit.Using (**degree * 2 * math.pi / 360) x1 = (type conversion int) (math.cos(radians) * xradius) + screensize/2** formula convert degree to radians.Using **pygame.draw.ellipse()** and **pygame.draw.circle() a** centre circle ,ellipse and another smaller circle on the ellipse are drawn.

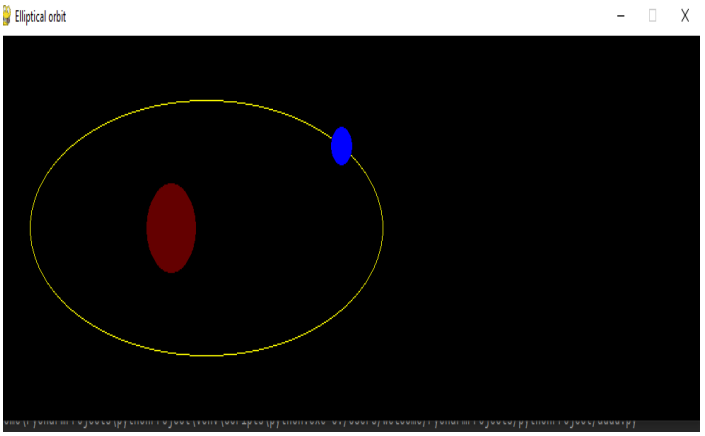
4. **A. Expected Input:**

   a. Load the ball image as input
   b. Entering X and Y as radius

5. B. Expected Output:

IV.

V.



6. Algorithm:

I.

Step 1: Start the program
Step 2: Set the screen size and background colour
Step 3: Set the speed of the moving ball
Step 4: Create a graphical window using set_mode()
Step 5: Set the caption
Step 6: Load the ball image and create a rectangle area covering the image.
Step 7: Use blit() method to copy the pixel color of the ball to the screen.
Step 8 : Set the background color of the screen and use the flip() method to make all the images visible
Step 9: Move the ball in specified speed.
Step 10: If the ball hits the edges of the screen reverse the direction.
Step 11: Create an infinite loop and repeat the steps 9 and 10 until the user exits the program.
Step 12: Stop the program

II.

Step 1: Start
Step 2: Set screen size and caption.
Step 3: Create clock variable
Step 4: Set the x and y radius of the ellipse.
Step 5: Starting from the degree 0 ending with 360 degrees in increments of 10 degrees calculate the (x1,y1) coordinates to find a point in the elliptical orbit.
Step 6: Convert degree to radiants (degree * 2*math.pi/360)
Step 7 : Set the background color ,draw the center circle ,ellipse and another smaller circle on the ellipse
Step 8: Refresh the screen for every 5 clock ticks.
Step 9: Repeat the steps 4 to 8 until user quits the program.
Step 10: Stop

7. Coding:

a. Write a program to develop a game activity like bouncing ball using Pygame

```
import pygame,sys
pygame.init()
size=width,height=1000,600
speed=[2,2]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("Bouncing ball")
ball=pygame.image.load("D:\IMAGES\Small ball.jpg")
ballrect=ball.get_rect()
while(True):
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
    ballrect=ballrect.move(speed)
    if ballrect.left<0 or ballrect.right>width:
        speed[0]=-speed[0]
    if ballrect.top<0  or ballrect.bottom>height:
        speed[1]=-speed[1]
    screen.fill("white")
    screen.blit(ball,ballrect)
    pygame.display.flip()
```

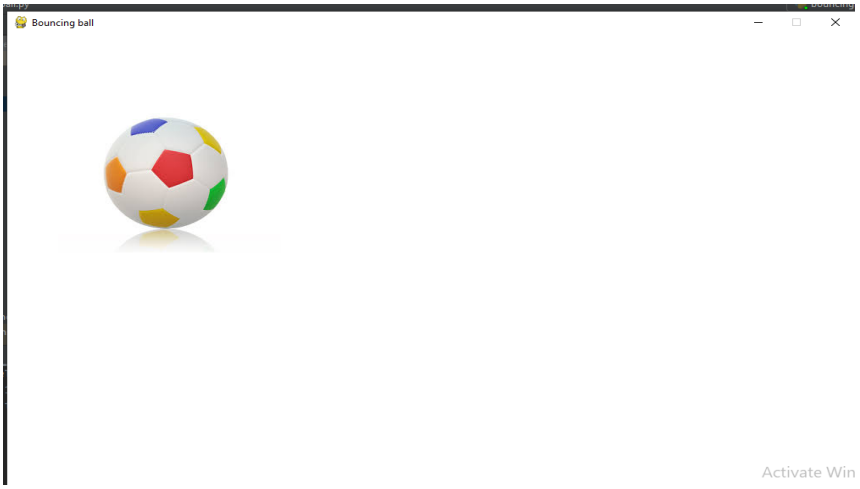
b. Write a program to develop a game activity like elliptical orbit using Pygame

```
import pygame
import math
import sys
pygame.init()
```

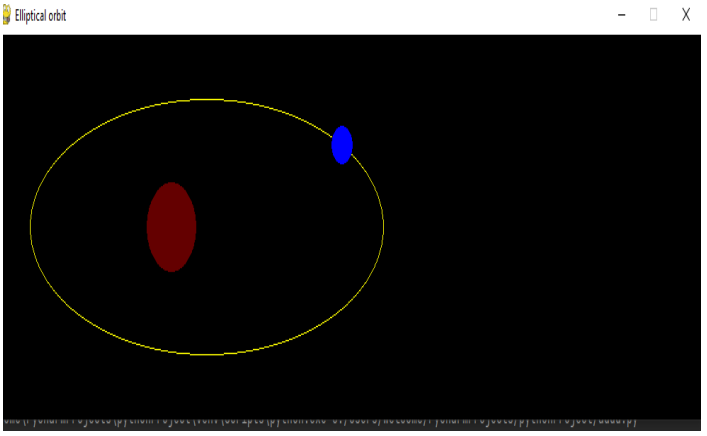
```
screen=pygame.display.set_mode((600,500))
pygame.display.set_caption("Elliptical Orbit")
clock=pygame.time.Clock()
while(True):
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
    xRadius=250
    yRadius=100
    for degree in range(0,360,10):
        x1=int(math.cos(degree*2*math.pi/360)*xRadius)+300
        y1=int(math.sin(degree*2*math.pi/360)*yRadius)+150
        screen.fill("black") #erase the screen by filling it with black color
        pygame.draw.circle(screen,(255,70,0),[300,150],40)
        pygame.draw.ellipse(screen,pygame.Color("yellow"),[50,50,500,200],1)
        pygame.draw.circle(screen,(0,255,0),[x1,y1],20)
        pygame.display.flip()
        clock.tick(5)
```

8. Output :

I.



II.



9. Result

Thus simulation of game activity like bouncing ball and elliptical orbit using pygame were successfully implemented and output was obtained.

10. Viva Questions

A. How do you make a circle surface in pygame?

To draw a circle in your pygame project you can **use draw**. circle() function.

B. Is pygame a GUI?

Pygame GUI is a module to help you make graphical user interfaces in for games written in pygame.

C. How to change screen background color in Pygame?

*** pygame.init():** This function is used to initialize all the pygame modules.
*** pygame.display.set_mode():** This function is used to initialize a screen for display.
*** fill():** This method is used to fill the display with the color specified.

D. How do you draw an ellipse in Pygame?

*Create a pygame. Surface object with a per-pixel alpha format and with the size of the shape.

*Draw the shape on the Surface.

*Rotate the Surface with the shape around its center.

* Blit the Surface with the shape onto the target Surface