LAB JOURNAL

Programming and Problem-Solving

Laboratory

Semester – II

A.Y 2020-2021

Group-II

(Div: A to F)

Name of Student : HARSHITA TOTALA
Roll Number: FE20IT0014
Seat No: F190080806
Batch: C1

Lab Coordinators :(Mrs. Vrushali U. Uttarwar)

CERTIFICATE

This is to certify that Harshita Totala of class F.E. batch C-1 Roll Number FE20IT014 has successfully completed this **Programming and Problem Solving Laboratory** practical assignments provided by **Mrs. Vrushali U. Uttarwar** during academic year **2020-2021 Semester-II** as per the guidelines

(Mrs. Vrushali U. Uttarwar)
Lab Teacher & Subject In-Charge

(Dr. Hnnie Williams)
Academic Coordinator

(Dr. S. K. Babar)
Dean Academics

INDEX

Sr. No	Title of Lab	Completion Date	Remark
1	Python Program to Calculate Salary of an employee	14/06/2021	
2	Python program to create list, find average, max, min of the list	21/06/2021	
3	Python program to calculate percentage and grade of a student	28/06/2021	
4	Python program to make a calculator to perform basic operations	01/07/2021	
5	Python program to accept list of N integers and partition list into two sub lists even and odd numbers	08/07/2021	
6	python program that accepts a string from user and perform string operations	13/07/2021	
7	Python program to create class employee and perform operations using it	14/07/2021	

I, Harshita Totala , Roll Number FE20IT014 have completed all the above lab assignments by myself, checked and submitted before deadline with Expected Attendance in SEM-II

Assignment No: -01

Class: - F.E. - C Division

Batch: -C1

Roll Number: FE20IT0014

Date of Assignment: - 11/06/2021

Date of Completion: - 14/06/2021

Grade: -

ASSIGNMENT. NO. 01

Title:

To calculate salary of an employee given his basic pay (take as input from user). Calculate salary of employee. Let HRA be 10 % of basic pay and TA be 5% of basic pay. Let employee pay professional tax as 2% of total salary. Calculate salary payable after deductions.

Objective:

Understand basic concepts of programming language and try to solve mathematical calculations in programming approach with the help of formulas and equations.

Problem Statement:

To Calculate salary payable of employee

Outcomes:

1. Students will be able to demonstrate calculations of employee salary.
2. Students will be able to demonstrate different Operator & formulas.
3. Students will be able to demonstrate different Operations on Available data of employee salary.

Hardware Requirement: Any CPU with i3 Processor or similar, 1 GB RAM or more,2 GB Hard Disk or more

Software Requirements: 32/64 bit Linux (Ubuntu/Fedora) Operating System, latest any Python Tool Like Anaconda,Pycharm, Python IDLE, Python Atom, Eclipse.

Theory:

Python Basics:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

What is Basic salary?

Basic salary is the base income of an individual. It is a fixed part of one's compensation package.

A basic salary depends on the employee's designation and also the industry in which the employee works.

Basic salary is the amount paid to an employee before any extras are added or taken off, such as reductions because of salary sacrifice schemes or an increase due to overtime or a bonus. Allowances, such as internet for home-based workers or contributions to phone usage, would also be added to the basic salary.Gross salary:

Gross salary is the amount calculated by adding up one's basic salary and allowances, before deduction of taxes and other deductions. It includes bonuses, over-time pay, holiday pay, and other differentials.

Gross Salary = Basic Salary + HRA + Other Allowances

Allowances:

An allowance is an amount received by the employee for meeting service requirements. Allowances are provided in addition to the basic salary and vary from company to company. Some common types of allowances are shown below:

- HRA or House Rent Allowance: It is an amount paid out to employees by companies for expenses related to rented accommodation.
- Leave Travel Allowance (LTA): LTA is the amount provided by the company to cover domestic travel expenses of an employee. It does not include the expenses for food, accommodation, etc. during the travel.
- Conveyance Allowance: This allowance is provided to employees to meet travel expenses from residence to work.
- Dearness Allowance: DA is a living allowance paid to employees to tackle the effects of inflation. It is applicable to government employees, public sector employees, and pensioners only.
- Other such allowances are the special allowance, medical allowance, incentives, etc.

HRA:

HRA received is not fully exempt from tax. HRA that you can claim is the lowest of the following:

- The total amount received as the HRA from the employer in the financial year.
- Actual rent paid in the year – 10% of the basic salary in the year.
- 50% of the annual basic salary if staying in a metro city or 40% of the annual basic salary if staying in a non-metro city.

Sample Example:

Python program to get employee wages and number of days worked from user and find Basic Pay, DA, HRA, PF and Net Pay.

(Note HRA, DA and PF are 10%,5%and 12% of basic pay respectively.)

Sample Input 1:

300

30

Sample Output 1:
Basic Pay:3000

DA: 150 HRA:300
PF:360
Net Pay: 3090

Algorithm:

0) Start
1) Accept the basic Pay

2)HRA=basic salary*(10/100)

3) TA=basic salary*(5/100)

4)Gross=basic salary+HRA+TA

5)Tax=Gross*(2/100)

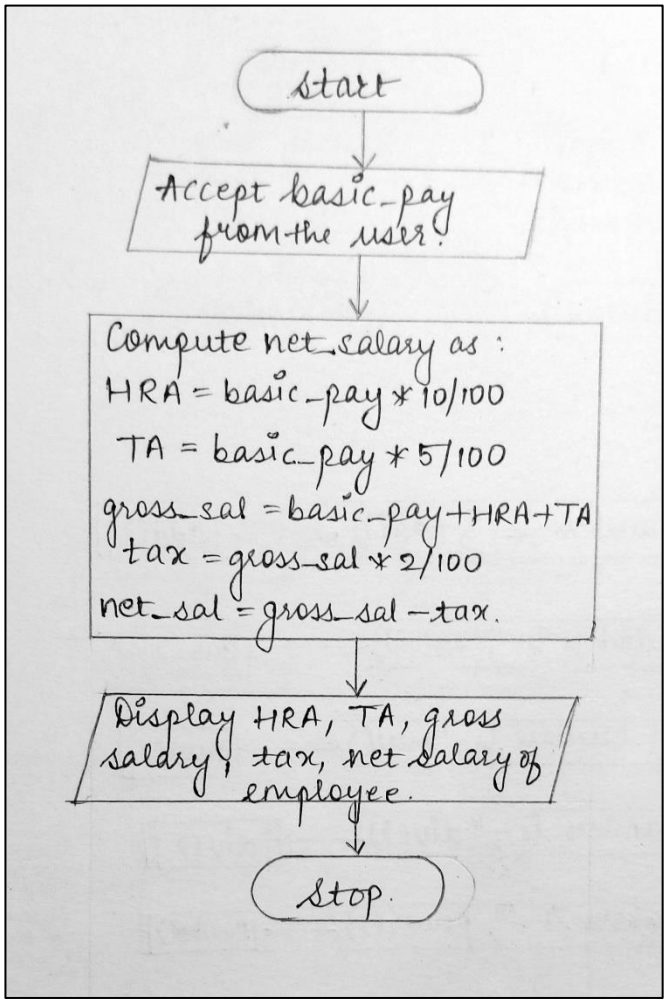
6)Net Salary=Gross-Tax

7) Display Net Payable Salary

8) Stop

STEP 1 : Start
STEP 2 : Accept basic_pay of the employee from the user.
STEP 3 : Calculate HRA using : HRA = basic_pay * (10/100)
STEP 4 : Calculate TA using : TA = basic_pay * 5/100
STEP 5 : Calculate Gross as : gross_sal = basic_pay + HRA + TA
STEP 6 : Calculate Tax as : tax = gross_sal * 2/100
STEP 7 : Calculate Net Salary as : net_sal = gross_sal - tax
STEP 8 : Display HRA, TA, gross salary, tax, net salary of employee.
STEP 9 : Stop

Flowchart:



Program:

```
File  Edit  Format  Run  Options  Window  Help
basic_pay=int(input("Enter the basic pay: "))

HRA=basic_pay*10/100

TA=basic_pay*5/100

gross_sal=basic_pay+HRA+TA

tax=gross_sal*2/100

net_sal=gross_sal-tax

print("BASIC SALARY- ",basic_pay)
print("HOUSE RENT ALLOWANCE- ",HRA)
print("TRAVELLING ALLOWANCE- ",TA)
print("GROSS SALARY OF THE EMPLOYEE-",gross_sal)
print("TAX PAID BY THE EMPLOYEE- ",tax)
print("NET SALARY OF AN EMPLOYEE- ",net_sal)
```

Output:

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/HP/Desktop/Harshita Phython/employee.py ==========
Enter the basic pay: 20000
BASIC SALARY-  20000
HOUSE RENT ALLOWANCE-  2000.0
TRAVELLING ALLOWANCE-  1000.0
GROSS SALARY OF THE EMPLOYEE- 23000.0
TAX PAID BY THE EMPLOYEE-  460.0
NET SALARY OF AN EMPLOYEE-  22540.0
>>>
```

Conclusion:

Thus, we have successfully understood concept of salary calculation formulas and performed salary calculation in python.

Assignment No: -02

Class: - F.E.-C Division

Batch: - C1

Roll Number: FE20IT014

Date of Assignment: - 18/06/2021

Date of Completion: - 21/06/2021

Grade-

Assignment No 02

Title:

To accept N numbers from user. Compute and display maximum in list, minimum in list, sum and average of numbers.

Objective:

Understand basic concepts of python programming language and try to solve list concepts related programs.

Problem Statement:

To Find maximum and minimum number, sum, average from given list.

Outcomes:

1. Students will be able to understand basic concept of list in python.

2. Students will be able to demonstrate different Operator & formulas on given list.

3. Students will be able to demonstrate different types of list examples easily.

Hardware Requirement: Any CPU with i3 Processor or similar, 1 GB RAM or more,2 GB Hard Disk or more

Software Requirements: 32/64 bit Linux(Ubuntu/Fedora)Operating System, latest any Python Tool Like Anaconda, Pycharm, Python IDLE, Python Atom, Eclipse.

Theory:

What is list?

Python offers a range of compound datatypes often referred to as sequences. List is one of the most frequently used and very versatile datatype used in Python.

How to create a list?

In Python programming, a list is created by placing all the items (elements) inside a square bracket [ ], separated by commas.
It can have any number of items and they may be of different types (integer, float, string etc

Algorithm:

1.Create an empty list named l

2.Read the value of n

3.Read the elements of the list until n

4.Assign l[0] as maxno

5.If l[i]>maxno then set maxno=l[i] 6.

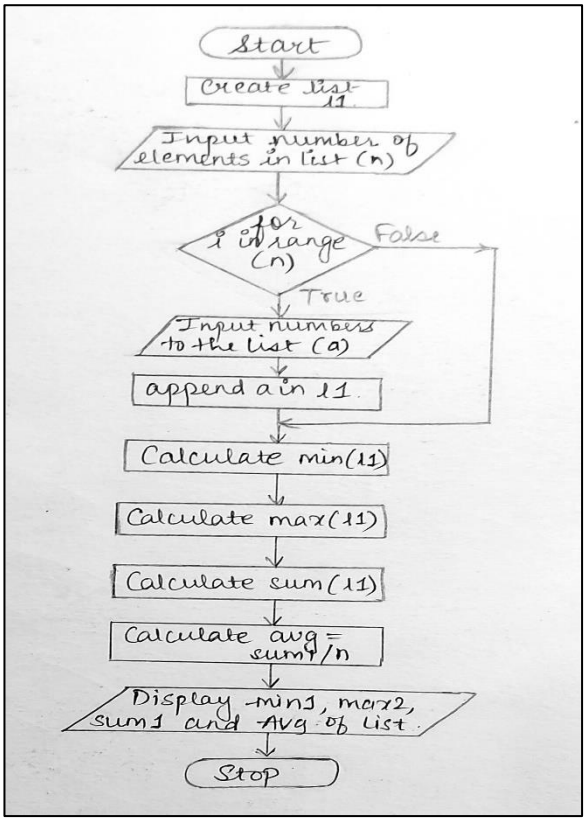
Increment i by 1

7.Repeat steps 5-6 until i<n

8.Print the value of maximum number

STEP 1: Start
STEP 2: Create empty list l1.
STEP 3: Input number of element into list.
STEP 4: Read the elements of the list till n using for loop.
STEP 5: Add the elements in list l1.
STEP 6: Find minimum number in the list using min() function.
STEP 7: Find maximum number in the list using max() function.
STEP 8: Find sum of the numbers using sum() function.
STEP 9: Calculate average of numbers by: Avg = sum/n.
STEP 10: Display the minimum, maximum, sum and average.
STEP 11: Stop.

Flowchart:



Program:

```
File  Edit  Format  Run  Options  Window  Help
l1=[]

n=int(input("Enter number of inputs to the list"))
for i in range(n):
    a=int(input("Enter a number to the list"))
    l1.append(a)

min1=min(l1)
max1=max(l1)
sum1=sum(l1)
Avg=sum1/n

print("Your Original List Is- ",l1)
print("Minimum Number In The List Is- ",min1)
print("Maximum Number In The  List Is- ",max1)
print("Sum Of  All Inputs In The List Is- ",sum1)
print("Average Of Inputs In The List Is- ",Avg)
```

Output:

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\Phython\max,min,sum of list.py ==========
=
Enter number of inputs to the list7
Enter a number to the list43
Enter a number to the list21
Enter a number to the list77
Enter a number to the list65
Enter a number to the list42
Enter a number to the list38
Enter a number to the list70
Your Original List Is-  [43, 21, 77, 65, 42, 38, 70]
Minimum Number In The List Is-  21
Maximum Number In The  List Is-  77
Sum Of  All Inputs In The List Is-  356
Average Of Inputs In The List Is-  50.857142857142854
>>>
```

Conclusion: Thus, we have studied and performed operation on list in python.

Assignment No: -03

Class: - F.E.-C Division

Batch: - C1

Roll Number: FE20IT014

Date of Assignment: - 25/06/2021

Date of Completion: - 28/06/2021

Grade-

Assignment No 03

Title:

To accept student's five courses marks and compute his/her result. Student is passing if he/she scores marks equal to and above 40 in each course. If student scores aggregate greater than 75%, then the grade is distinction. If aggregate is 60>= and <75 then the grade if first division. If aggregate is 50>= and <60, then the grade is second division. If aggregate is 40>= and <50, then the grade is third division

Objective:

Understand basic concepts of programming language and try to solve mathematical calculations in programming approach with the help of formulas and equations.

Problem Statement:

To calculate percentage,aggregate of given five subjects in python.

Outcomes:

1. Students will be able to understand basic concept of percentage calculation in python.
2. Students will be able to demonstrate different Operator & formulas for grade and sum.
3. Students will be able to demonstrate different Operations by using own logic.

Hardware Requirement: Any CPU with i3 Processor or similar, 1 GB RAM or more,2 GB Hard Disk or more

Software Requirements: 32/64 bit Linux(Ubuntu/Fedora)Operating System, latest any Python Tool Like Anaconda, Pycharm, Python IDLE, Python Atom, Eclipse.

Theory:

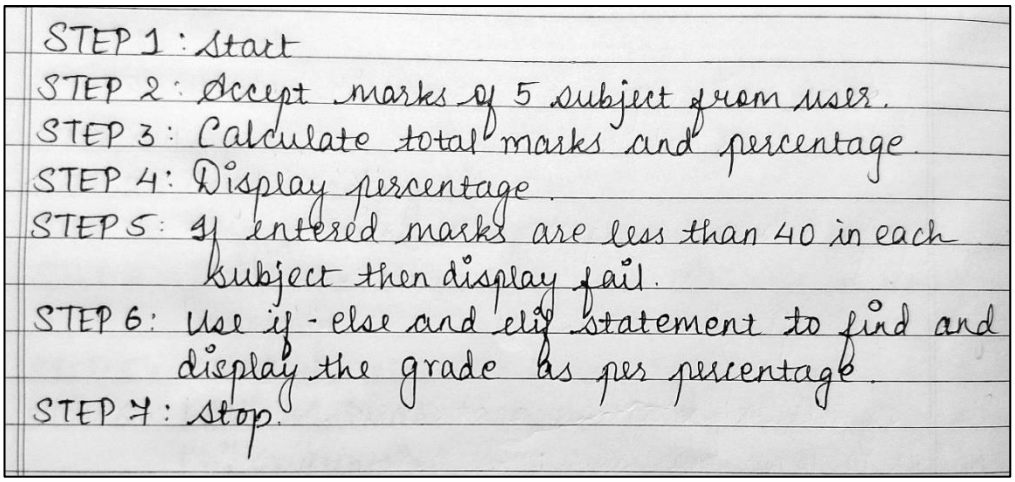
In this assignment user has to enter five different marks for five subjects. Next, it will find the Total, average, and Percentage of those Five Subjects. For this, we are using the arithmetic operator to perform arithmetic operations.

Problem Description

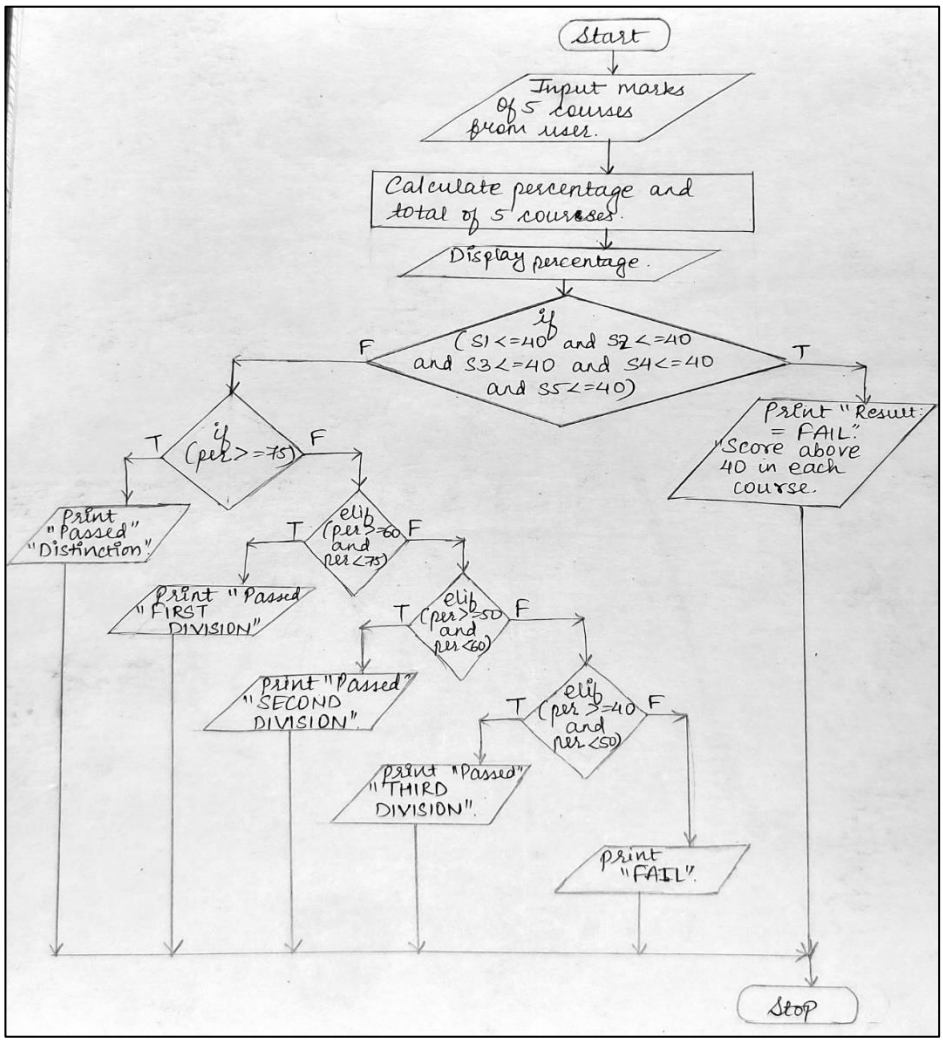
The program takes in the marks of 5 subjects and displays the grade.

Algorithm

1. Take in the marks of 5 subjects from the user and store it in different variables.
2. Find the average of the marks.
3. Use an else condition to decide the grade based on the average of the marks.
4. Exit.



Flowchart:



Program/Source Code:

```
File  Edit  Format  Run  Options  Window  Help
S1=int(input("Enter marks in English subject-"))
S2=int(input("Enter marks in Hindi subject-"))
S3=int(input("Enter marks in Science  subject-"))
S4=int(input("Enter marks in Maths  subject-"))
S5=int(input("Enter marks in Computer subject-"))

total=S1+S2+S3+S4+S5
per=total/500*100
print("Your Percentage is- ",per,"%")

if(S1<=40 and S2<=40 and S3<=40 and S4<=40 and S5<=40):
    print("FAIL! You have to score above 40 in each subject.")
else:
    if(per>=75):
        print("Result-PASS \nGrade-'DISTINCTION' ")
    elif(per>=60 and per<75):
         print("Result-PASS \nGrade-'FIRST DIVISION' ")
    elif(per>=50 and per<65):
         print("Result-PASS \nGrade-'SECOND DIVISION' ")
    elif(per>=40 and per<50):
         print("Result-PASS \nGrade-'THIRD DIVISION' ")
    else:
        print("Result-FAIL")
        print("Better luck next time")

print("----ALL THE BEST FOR YOUR FUTURE !----")
```

Output:

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/HP/Desktop/Phython/percentage and grade.py ==========
=
Enter marks in English subject-67
Enter marks in Hindi subject-78
Enter marks in Science  subject-65
Enter marks in Maths  subject-90
Enter marks in Computer subject-92
Your Percentage is-  78.4 %
Result-PASS
Grade-'DISTINCTION'
----ALL THE BEST FOR YOUR FUTURE !----
>>>
```

Conclusion: Thus, we have performed how to calculate percentage and grade, average in python.

Assignment No: -04

Class: - F.E.-C Division

Batch: - C1

Roll Number: FE20IT014

Date of Assignment: 29/06/2021

Date of Completion: - 01/07/2021

Grade-

Assignment No -04

Title:

To accept the number and Compute a) square root of number, b) Square of number, c) Cube of number d) check for prime, e) factorial of number

Objective:

Understand basic concepts of programming language and try to solve mathematical calculations in programming approach with the help of formulas and equations.

Problem Statement:

To perform number different operation on given number in python.

Outcomes:

1. Students will be able to understand basic concept of number operations logic.

2. Students will be able to demonstrate different integer number for identifying number operations.

3. Students will be able to demonstrate different Operations on any integer by using own logic.

Hardware Requirement: Any CPU with i3 Processor or similar, 1 GB RAM or more,2 GB Hard Disk or more

Software Requirements: 32/64 bit Linux(Ubuntu/Fedora)Operating System, latest any Python Tool Like Anaconda,Pycharm, Python IDLE, Python Atom, Eclipse.

Theory:

To understand this experiment, you should have the knowledge of following Python programming topics:

Python Input, Output and Import

Python provides numerous built-in functions that are readily available to us at the Python prompt.

Some of the functions like input() and print() are widely used for standard input and output operations respectively. Let us see the output section first.

Python Output Using print() function

We use the print() function to output data to the standard output device (screen).

We can also output data to a file, but this will be discussed later. An example use is given below.

```
print('This sentence is output to the screen')

# Output: This sentence is output to the screen
```

```
a = 5
print('The value of a is', a)
# Output: The value of a is 5
```

Python Data Types

Every value in Python has a datatype. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

There are various data types in Python. Some of the important types are listed below.

Python Numbers

Integers, floating point numbers and complex numbers falls under Python numbers category. They are defined as int, float and complex class in Python.

We can use the type() function to know which class a variable or a value belongs to and the isinstance() function to check if an object belongs to a particular class.

```
a = 5
print(a, "is of type", type(a))
a = 2.0
print(a, "is of type", type(a))
a = 1+2j
print(a, "is complex number?", isinstance(1+2j,complex))
```

Python Operators

Operators are special symbols in Python that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.

For example:

```
>>> 2+35
```

Here, + is the operator that performs addition. 2 and 3 are the operands and 5 is the output of the operation.

a) **Python math function | sqrt()**

sqrt() function is an inbuilt function in Python programming language that returns the square root of any number.

Syntax:

math.sqrt(x)

Parameter:
x is any number such that x>=0

Returns:

It returns the square root of the number passed in the parameter.

Example:

```
# Python3 program to demonstrate the
# sqrt() method
# import the math module
import math
# print the square root of 0
print(math.sqrt(0))
# print the square root of 4
print(math.sqrt(4))
# print the square root of 3.5
print(math.sqrt(3.5))
```

b) Square of a Number

In this article, we will show you, How to write a Python Program to Calculate Square of a Number using Arithmetic Operators, and Functions with example.

allows the user to enter any numerical value. Next, it will finds the square of that number using Arithmetic Operator.

Example:

```
# Python Program to Calculate Square of a Number number
= float(input(" Please Enter any numeric Value : ")) square
= number * number
print("The Square of a Given Number {0} = {1}".format(number, square))
```

c) Cube of a Number

we will show you, How to write a Python Program to Calculate Cube of a Number using Arithmetic Operators, and Functions with example.

Example:

```
# Python Program to Calculate Cube of a Number
number = float(input(" Please Enter any numeric Value : "))
cube = number * number * number
print("The Cube of a Given Number {0} = {1}".format(number, cube))
```

d) Prime Number:

A positive integer greater than 1 which has no other factors except 1 and the number itself is called a prime number. 2, 3, 5, 7 etc. are prime numbers as they do not have any other factors. But 6 is not prime (it is composite) since, 2 x 3 = 6.

Given a positive integer N. The task is to write a Python program to check if the number is prime or not.

Definition: A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. The first few prime numbers are {2, 3, 5, 7, 11, ….}.

Examples :

nput: n = 11
Output: true

Input: n = 15
Output: false

Input: n = 1
Output: false

The idea to solve this problem is to iterate through all the numbers starting from 2 to (N/2) using a for loop and for every number check if it divides N. If we find any number that divides, we return false. If we did not find any number between 2 and N/2 which divides N then it means that N is prime and we will return True.

Below is the Python program to check if a number is prime:

```
# Python program to check if #
given number is prime or not

num = 11
# If given number is greater than 1
if num > 1:

   # Iterate from 2 to n / 2
   for i in range(2, num//2):

      # If num is divisible by any number between
      # 2 and n / 2, it is not prime
      if (num % i) == 0:
         print(num, "is not a prime number")
         break
   else:
      print(num, "is a prime number")
else:
   print(num, "is not a prime number")
```

e) Factorial Number:

The factorial of a number is the product of all the integers from 1 to that number.

For example, the factorial of 6 (denoted as 6!) is 1*2*3*4*5*6 = 720. Factorial is not defined for negative numbers and the factorial of zero is one, 0! = 1.

factorial() in Python

Not many people know, but python offers a direct function that can compute the factorial of a number without writing the whole code for computing factorial.

Naive method to compute factorial

```
# Python code to demonstrate naive method # to compute factorial
n = 23
fact = 1
for i in range(1,n+1): fact = fact * i
print ("The factorial of 23 is : ",end="")
print (fact)
```

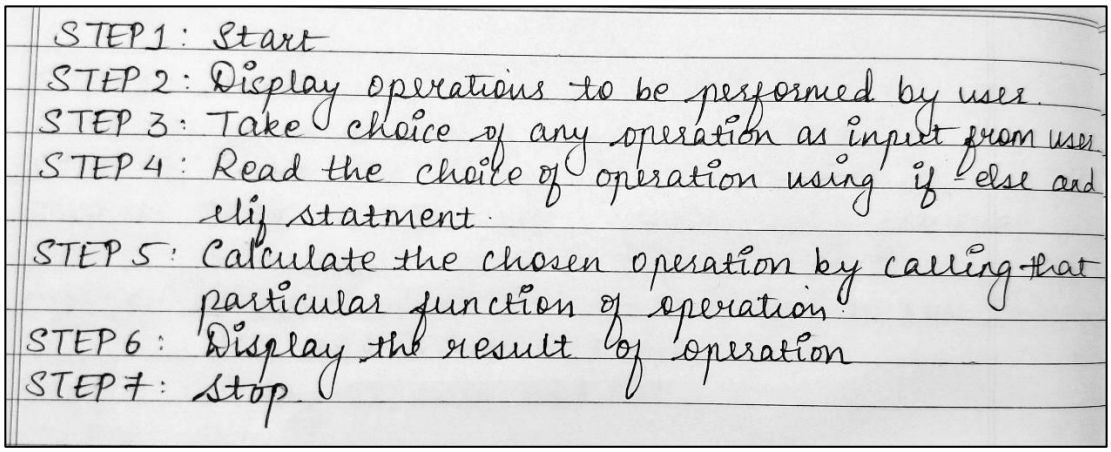
Using math.factorial()

This method is defined in “math” module of python. Because it has C type internal implementation, it is fast.

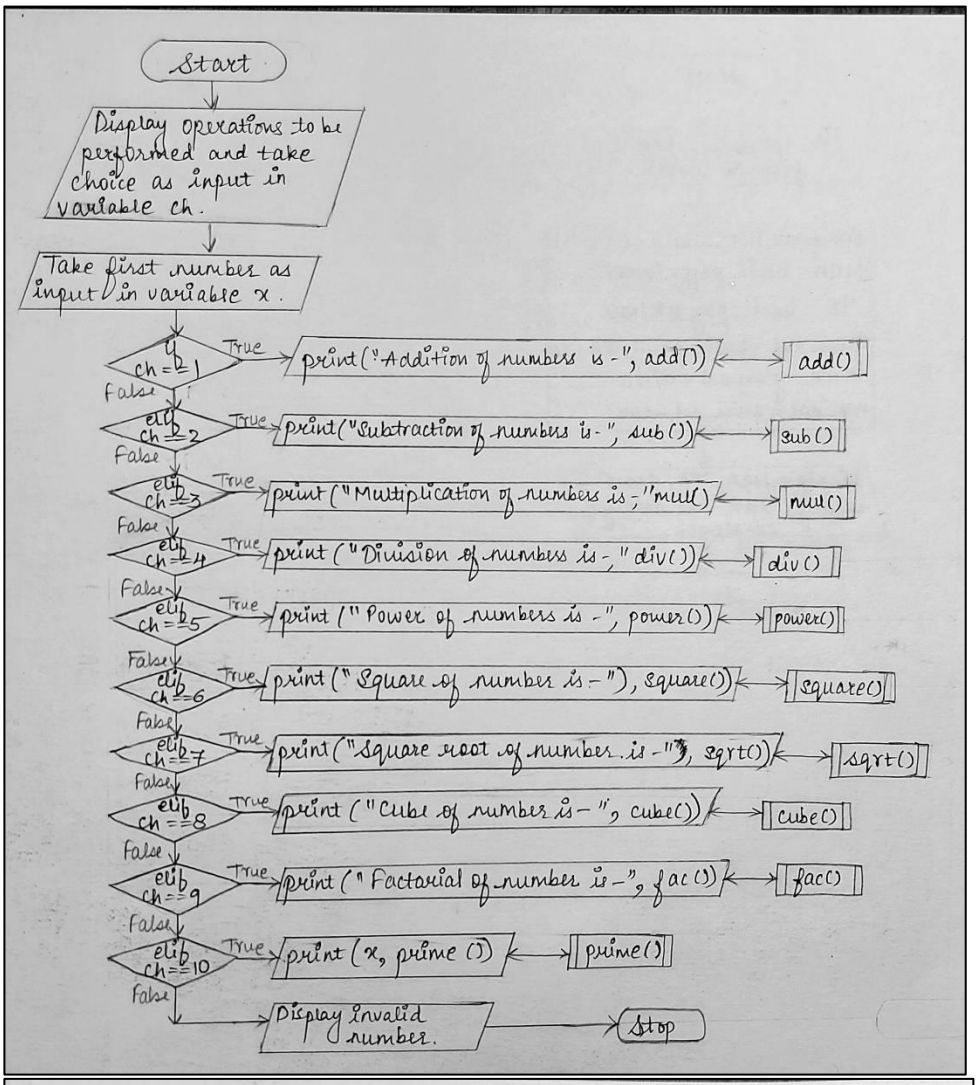
math.factorial(x) Parameters :
x : The number whose factorial has to be computed.
Return value :
Returns the factorial of desired number.
Exceptions:
Raises Value error if number is negative or non-integral.

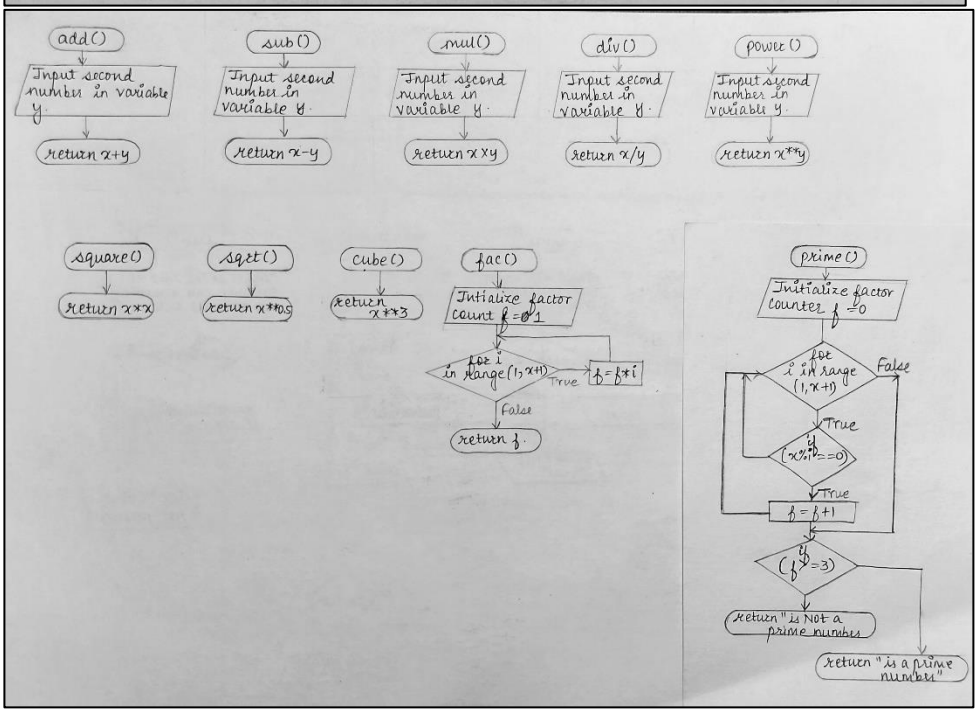
```
# Python code to demonstrate math.factorial() import math

print ("The factorial of 23 is : ", end="")
print (math.factorial (23))
```

Algorithm:



Flowchart: Draw Separate flowchart for a) ,
b) , c), d) ,e)





Program:

File Edit Format Run Options Window Help

```
def add():
    y=int(input("Enter second  number- "))
    return x+y

def sub():
    y=int(input("Enter second  number- "))
    return x-y

def mul():
    y=int(input("Enter second number- "))
    return x*y

def div():
    y=int(input("Enter second  number- "))
    return x/y

def power():
    y=int(input("Enter second  number- "))
    return x**y

def square():
    return x*x

def sqrt():
    return x**0.5

def cube():
    return x**3

def fac():
    f=1
    for i in range(1,x+1):
        f=f*i
    return f
def prime():
    f=0
    for i in range(1,x+1):
        if(x%i==0):
            f=f+1
    if(f>=3):
        return " is Not a  Prime Number"
    else:
        return " is  a Prime Number"

print("---OPERATIONS---\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n5.Power\n6.Square of Number"
      "\n7.Sqaure Root\n8.Cube of Number\n9.Factorial of Number\n10.Check of Prime Number \n")

ch=int(input("Select any operation to perform- "))
x=int(input("Enter first number- "))

if(ch==1):
     print("ADDITION OF NUMBERS IS- ",add())
elif(ch==2):
    print("SUBTRACTION OF NUMBERS IS- ",sub())
elif(ch==3):
    print("MULTIPLICATION OF NUMBERS IS- ",mul())
elif(ch==4):
    print("DIVISION OF NUMBERS IS- ",div())
elif(ch==5):
    print(" POWER  OF NUMBERS IS - ",power())
elif(ch==6):
    print("SQUARE OF NUMBER IS- ",square())
elif(ch==7):
    print("SQUARE ROOT  OF NUMBER IS- ",sqrt())
elif(ch==8):
    print("CUBE  OF NUMBER IS- ",cube())
elif(ch==9):
    print("FACTORIAL OF NUMBER IS- ",fac())
elif(ch==10):
    print(x,prime())
else:
    print("INVALID NUMBER!")
```

Output :

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\HP\Desktop\Phython\calculator.py ==============
---OPERATIONS---
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Power
6.Square of Number
7.Sqaure Root
8.Cube of Number
9.Factorial of Number
10.Check of Prime Number

Select any operation to perform- 10
Enter first number- 7
7  is  a Prime Number
>>>
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\HP\Desktop\Phython\calculator.py ==============
---OPERATIONS---
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Power
6.Square of Number
7.Sqaure Root
8.Cube of Number
9.Factorial of Number
10.Check of Prime Number

Select any operation to perform- 9
Enter first number- 6
FACTORIAL OF NUMBER IS-  720
>>>
```

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\HP\Desktop\Phython\calculator.py ==============
---OPERATIONS---
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Power
6.Square of Number
7.Sqaure Root
8.Cube of Number
9.Factorial of Number
10.Check of Prime Number

Select any operation to perform- 5
Enter first number- 6
Enter second  number- 3
 POWER  OF NUMBERS IS -  216
>>>
```

Conclusion: Thus,in this experiment we have studied and performed mathematical operation on given number.

Assignment No: -05

Class: - F.E.-C Division

Batch: - C1

Roll Number: FE20IT014

Date of Assignment: - 05/07/2021

Date of Completion: - 08/07/2021

Grade-

Assignment No 05

Title:

To accept list of N integers and partition list into two sub lists even and odd numbers.

Objective:

Understand basic concepts of programming language and try to solve mathematical calculations in programming approach with the help of formulas and equations.

Problem Statement:

To perform odd and even from two different list in python.

Outcomes:

1. Students will be able to understand basic concept of even odd logic.
2. Students will be able to demonstrate list operations.
3. Students will be able to demonstrate different Operations on any integer by using own logic.

Hardware Requirement: Any CPU with i3 Processor or similar, 1 GB RAM or more,2 GB Hard Disk or more

Software Requirements: 32/64 bit Linux(Ubuntu/Fedora)Operating System, latest any Python Tool Like Anaconda, PyCharm, Python IDLE, Python Atom, Eclipse.

Theory:

Every value in Python has a datatype. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements. Each element or value that is inside of a list is called an item. Just as strings are defined as characters between quotes, lists are defined by having values between square brackets [ ].

Lists are great to use when you want to work with many related values. They enable you to keep data together that belongs together, condense your code, and perform the same methods and operations on multiple values at once.

When thinking about Python lists and other data structures that are types of collections, it is useful to consider all the different collections you have on your computer: your assortment of files, your song playlists, your browser bookmarks, your emails, the collection of videos you can access on a streaming service, and more.

List is an ordered sequence of items. It is one of the most used datatype in Python and is very flexible. All the items in a list do not need to be of the same type.

Declaring a list is pretty straight forward. Items separated by commas are enclosed within brackets [

].
>>> a = [1, 2.2, 'python']

We can use the slicing operator [ ] to extract an item or a range of items from a list. Index starts form 0 in Python.

Example

```
a = [5,10,15,20,25,30,35,40] # a[2] = ?
print("a[2] = ", a[2]) # a[0:3] = ?
print("a[0:3] = ", a[0:3]) # a[5:] =
print("a[5:] = ", a[5:])
```

Given a list of numbers, write a Python program to count Even and Odd numbers in a List.

Example:

Input: list1 = [2, 7, 5, 64, 14] Output: Even = 3, odd = 2

Input: list2 = [12, 14, 95, 3] Output: Even = 2, odd = 2

Example 1: count Even and Odd numbers from given list using for loop

Iterate each element in the list using for loop and check if num % 2 == 0, the condition to check even numbers. If the condition satisfies, then increase even count else increase odd count.

Algorithm:

1. Take in the number of elements and store it in a variable.
2. Take in the elements of the list one by one.
3. Use a for loop to traverse through the elements of the list and an if statement to check if the element is even or odd.
4. If the element is even, append it to a separate list and if it is odd, append it to a different one.
5. Display the elements in both the lists.
6. Exit.

Program Explanation

1. User must enter the number of elements and store it in a variable.
2. User must then enter the elements of the list one by one using a for loop and store it in a list.
3. Another for loop is used to traverse through the elements of the list.
4. The if statement checks if the element is even or odd and appends them to separate lists.
5. Both the lists are printed.

Runtime Test Cases Case 1:

```
Enter number of elements:5
Enter element:67 Enter element:43 Enter element:44 Enter element:22 Enter element:455 The even list [44, 22]
The odd list [67, 43, 455]
```

Case 2:

Enter number of elements:3 Enter element:23

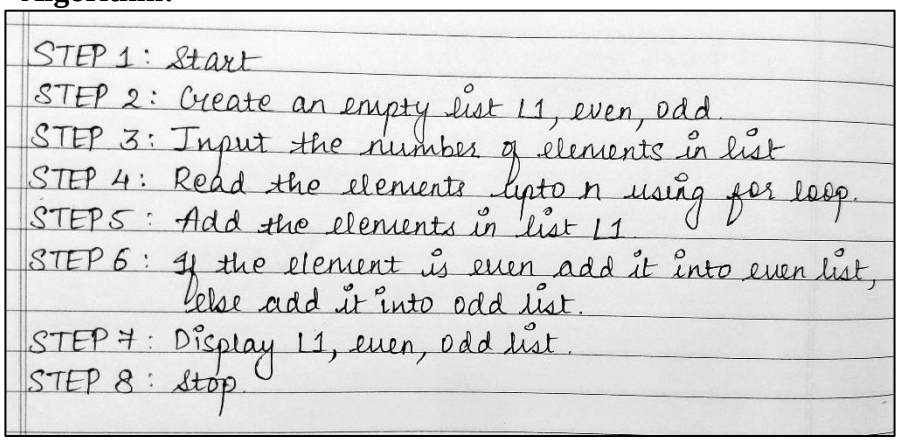
Enter element:44

Enter element:99 The even list [44] The odd list [23, 99]

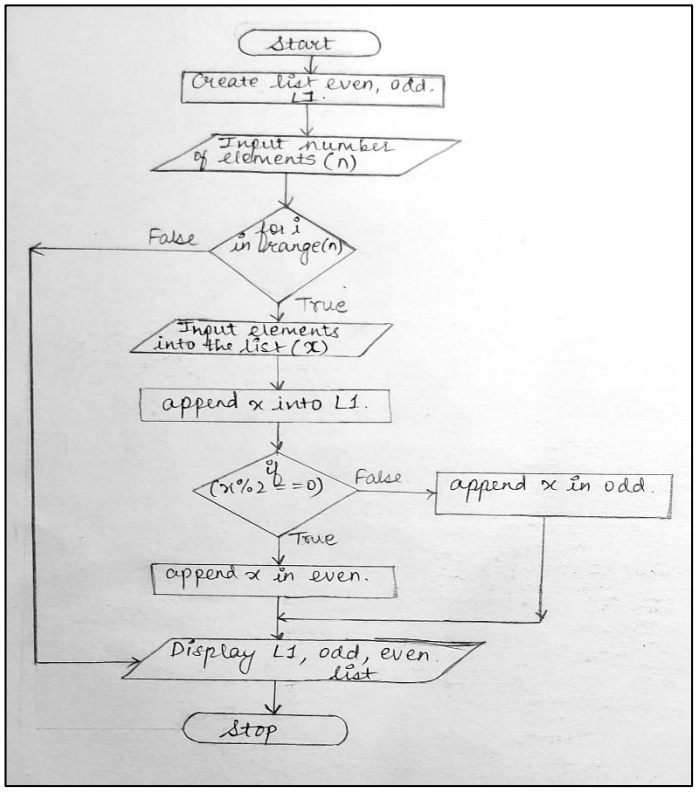
Program/Source Code :

```
a=[]
n=int(input("Enter number of elements:")) for i in range(1,n+1):
b=int(input("Enter element:")) a.append(b)
even=[] odd=[] for j in a:
    if(j%2==0):
        even.append(j) else:
odd.append(j) print("The even list",even) print("The odd list",odd)
```

Algorithm:



Flowchart:



Program:

File Edit Format Run Options Window Help

```
even=[]
odd=[]
L1=[]
n=int(input("Enter number of elements-"))

for i in range(n):
    x=int(input("Enter elements in list-"))
    L1.append(x)

    if(x%2==0):
        even.append(x)
    else:
        odd.append(x)

print("Original List-",L1)
print("Even List-",even)
print("Odd List-",odd)
```

Output:

```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\HP\Desktop\Phython\List of even and odd.py ==========
Enter number of elements-6
Enter elements in list-21
Enter elements in list-45
Enter elements in list-67
Enter elements in list-44
Enter elements in list-90
Enter elements in list-82
Original List- [21, 45, 67, 44, 90, 82]
Even List- [44, 90, 82]
Odd List- [21, 45, 67]
>>>
```

Conclusion:

Thus, in this experiment we have studied

Assignment No: -06

Class: - F.E.-C Division

Batch: - C1

Roll Number: FE20IT014

Date of Assignment: - 08/07/2021

Date of Completion: - 13/07/2021

Grade : -

Assignment No 06

Title:

Write a python program that accepts a string from user and perform following string operations- i. Calculate length of string ii. String reversal iii. Equality check of two strings iii. Check palindrome ii. Check substring

Objective:

Understand basic concepts of programming language and try to solve mathematical calculations in programming approach with the help of formulas and equations.

Problem Statement:

To perform String operation in python**.**

Outcomes:

1. Students will be able to understand basic concept of string in python.

2. Students will be able to demonstrate different operations on string.

3. Students will be able to demonstrate different Operations on any string by using own logic.

Hardware Requirement: Any CPU with i3 Processor or similar, 1 GB RAM or more,2 GB Hard Disk or more

Software Requirements: 32/64 bit Linux(Ubuntu/Fedora)Operating System, latest any Python Tool Like Anaconda,Pycharm, Python IDLE, Python Atom, Eclipse.

Theory:

A string is a list of characters in order. A character is anything you can type on the keyboard in one keystroke, like a letter, a number, or a backslash. Strings can have spaces: "hello world". An empty string is a string that has 0 characters.

A string is a list of characters in order.
A character is anything you can type on the keyboard in one keystroke,
like a letter, a number, or a backslash.

Strings can have spaces: "hello world".

An empty string is a string that has 0 characters.

Python strings are immutable

Python recognize as strings everything that is delimited by quotation marks (" " or ' ').

String Manipulation

To manipulate strings, we can use some of Pythons built-in methods.

```
word = "Hello World"

>>> print word
Hello World
```

Accessing

Use [ ] to access characters in a string

```
word = "Hello World"

letter=word[0]

>>> print letter
H
```

Length

```
word = "Hello World"

>>> len(word)
11
```

Finding

```
word = "Hello World"

>>> print word.count('l')       # count how many times l is in the string
3

>>> print word.find("H")        # find the word H in the string
0

>>> print word.index("World") # find the letters World in the string
6
```

Count

```
s =  "Count, the number     of spaces"

>>> print s.count('
') 8
```

Slicing

Use [# : #] to get set of letter

Keep in mind that python, as many other languages, starts to count from 0!!

word = "Hello World"

print word[0] #get one char of the word
print word[0:1] #get one char of the word
(same as above) print word[0:3] #get the first
three char
print word[:3] #get the
first three char print word[-3:]
#get the
last three char
print word[3:] #get all but the three
first char print word[:-3] #get all but the
three last character

word = "Hello World"

word[start:end] # items start through
end-1 word[start:] # items start through the
rest of the list
word[:end] # items from the beginning
through end-1 word[:] # a copy of the whole
list

Split Strings

word = "Hello World"

>>> word.split(' ') # Split on whitespace ['Hello', 'World']

Startswith / Endswith word = "hello world"

>>> word.startswith("H") True

>>> word.endswith("d") True

>>> word.endswith("w") False
Changing Upper and Lower Case Strings string = "Hello World"

>>> print string.upper()

HELLO WORLD

>>> print string.lower() hello world

>>> print string.title() Hello World

>>> print string.capitalize() Hello world

>>> print string.swapcase() hELLO wORLD

Reverse words in a given string

Example: Let the input string be “i like this program very much”. The function should change the string to “much very program this like i”

Input: hello Output: olleh

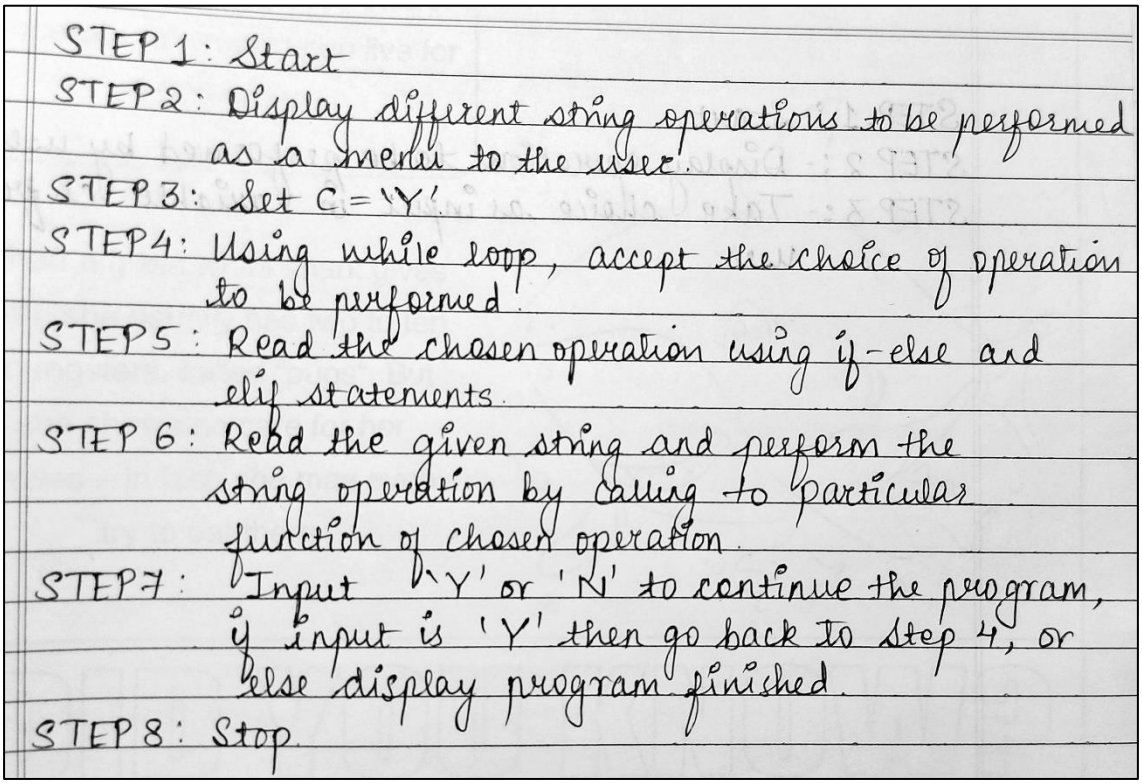
The format() Method for Formatting Strings

The format() method that is available with the string object is very versatile and powerful in formatting strings. Format strings contains curly braces {} as placeholders or replacement fields which gets replaced.

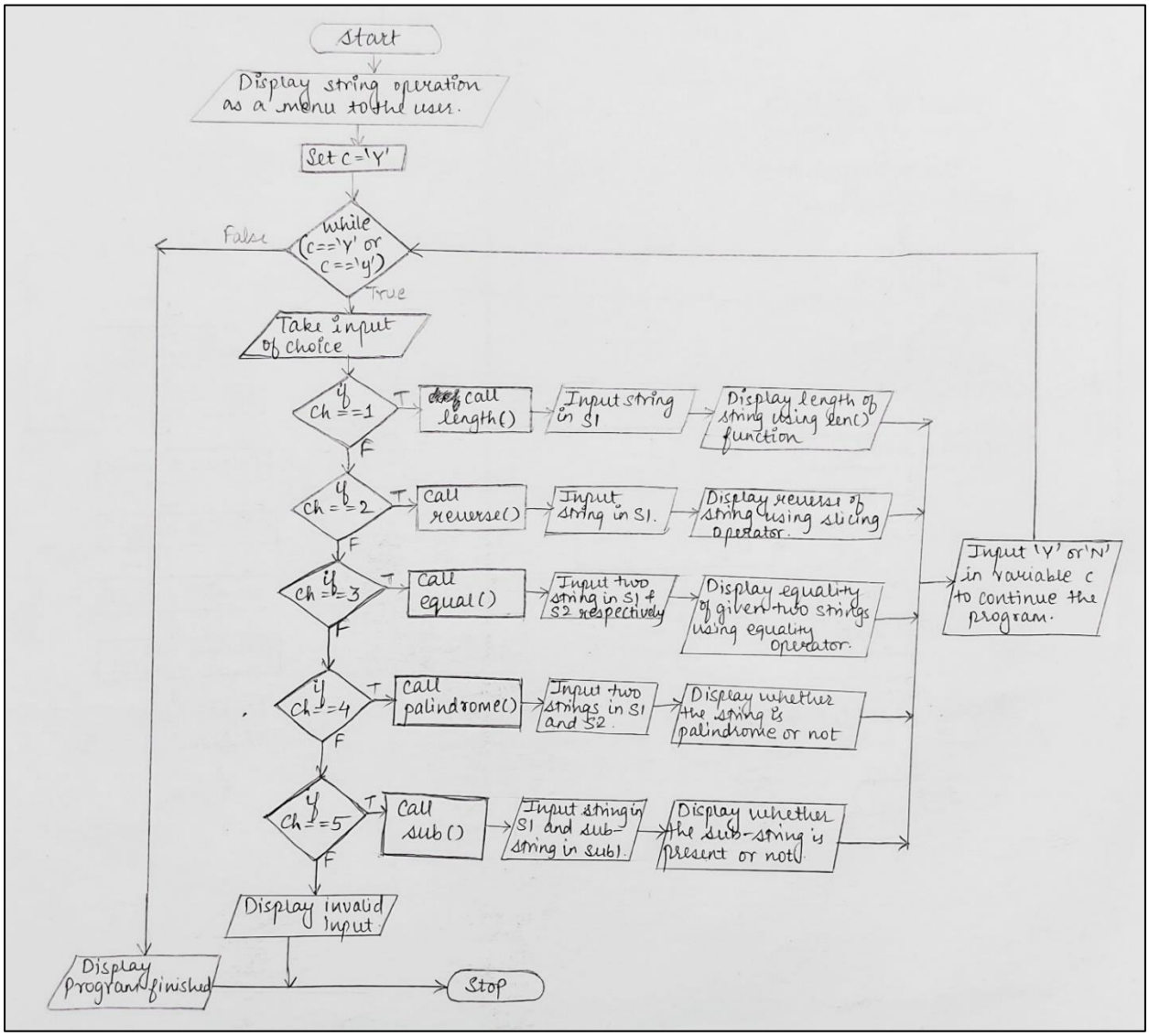
We can use positional arguments or keyword arguments to specify the order.

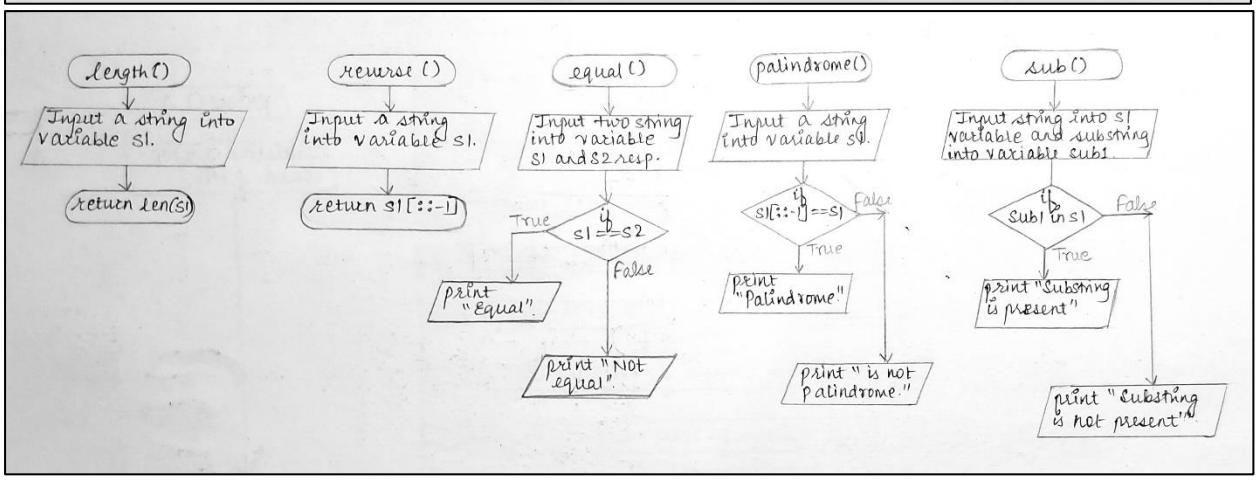
The format() method can have optional format specifications. They are separated from field name using colon. For example, we can left-justify <, right-justify >or center ^ a string in the given space. We can also format integers as binary, hexadecimal etc. and floats can be rounded or displayed in the exponent format. There are a ton of formatting you can use.

Algorithm:



Flowchart:


Start
Display string operation as a menu to the user.
Set c='Y'
while (c=='Y' or c=='y')
False
True
Take input of choice
if Ch==1
T
call length()
Input string in S1
Display length of string using len() function
F
if Ch==2
call reverse()
Input string in S1.
Display reverse of string using slicing operator.
if Ch==3
call equal()
Input two string in S1 & S2 respectively
Display equality of given two strings using equality operator.
if Ch==4
call palindrome()
Input two strings in S1 and S2.
Display whether the string is palindrome or not
if Ch==5
call sub()
Input string in S1 and sub-string in sub1.
Display whether the sub-string is present or not
Input 'Y' or 'N' in variable c to continue the program.
Display invalid input
Display Program finished
Stop


length()
Input a string into variable s1.
return len(s1)
reverse()
Input a string into variable s1.
return s1[::-1]
equal()
Input two string into variable s1 and s2 resp.
if s1==s2
True
False
print "Equal".
print "Not equal".
palindrome()
Input a string into variable s1.
if s1[::-1]==s1
False
True
print "Palindrome"
print " is not palindrome."
sub()
Input string into s1 variable and substring into variable sub1.
if sub1 in s1
False
True
print "Substring is present"
print "Substring is not present"

Program :

File Edit Format Run Options Window Help

```
print("----MENU---- \n1.Length Of The String\n2.String Reversal\n3.Equality Check Of Two String\n"
      "4.Check Palindrome\n5.Check For Sub-String")
def length():
    s1 = input("Enter A String-")
    print("Length of String", s1, "Is", len(s1))

def reverse():
    s1= input("Enter A String-")
    print("Reverse of",s1,"Is",s1[::-1])

def equal():
    s1= input("Enter A String-")
    s2= input("Enter A String-")
    if(s1==s2):
        print(s1,"Is Equal",s2)
    else:
        print(s1,"Is Not Equal",s2)

def palindrome():
    s1= input("Enter A String-")
    if(s1[::-1]==s1):
        print(s1,"Is Palindrome")
    else:
        print(s1,"Is Not Palindrome")

def sub():
    s1= input("Enter A String-")
    sub1= input("Enter Sub-String -")
    if(sub1 in s1):
        print(sub1,"Is Found In",s1)
    else:
        print(sub1,"Is Not Found In",s1)
c='Y'
while(c=='Y' or c=='y'):
    ch=int(input("\n Enter Your Choice:-"))
    if(ch==1):
        length()
    elif(ch==2):
        reverse()
    elif(ch==3):
        equal()
    elif(ch==4):
        palindrome()
    elif(ch==5):
        sub()
    else:
        print("Invalid Input!")
    c=input("Do You Want To Continue? Yes(Y)/No(N) ")
print("*Program Finished*")
```

Output :

```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\HP\Desktop\Phython\Strings.py ================
----MENU----
1.Length Of The String
2.String Reversal
3.Equality Check Of Two String
4.Check Palindrome
5.Check For Sub-String

 Enter Your Choice:-1
Enter A String-Welcome Home
Length of String Welcome Home Is 12
Do You Want To Continue? Yes(Y)/No(N) Y

 Enter Your Choice:-6
Invalid Input!
Do You Want To Continue? Yes(Y)/No(N) Y

 Enter Your Choice:-5
Enter A String-Priya loves cooking
Enter Sub-String -loves
loves Is Found In Priya loves cooking
Do You Want To Continue? Yes(Y)/No(N) Y

 Enter Your Choice:-2
Enter A String-program
Reverse of program Is margorp
Do You Want To Continue? Yes(Y)/No(N) y

 Enter Your Choice:-3
Enter A String-Computer
Enter A String-computer
Computer Is Not Equal computer
Do You Want To Continue? Yes(Y)/No(N) y

 Enter Your Choice:-4
Enter A String-level
level Is Palindrome
Do You Want To Continue? Yes(Y)/No(N) n
*Program Finished*
>>>
```

Conclusion: Thus, in this assignment we have studied and performed string operation successfully.

Assignment No: -07

Class: - F.E.-C Division

Batch: - C1

Roll Number: FE20IT014

Date of Assignment: - 09/07/2021

Date of Completion: - 14/07/2021

Grade : -

Assignment No :07

Aim : To gain the concepts of Object oriented programming in python

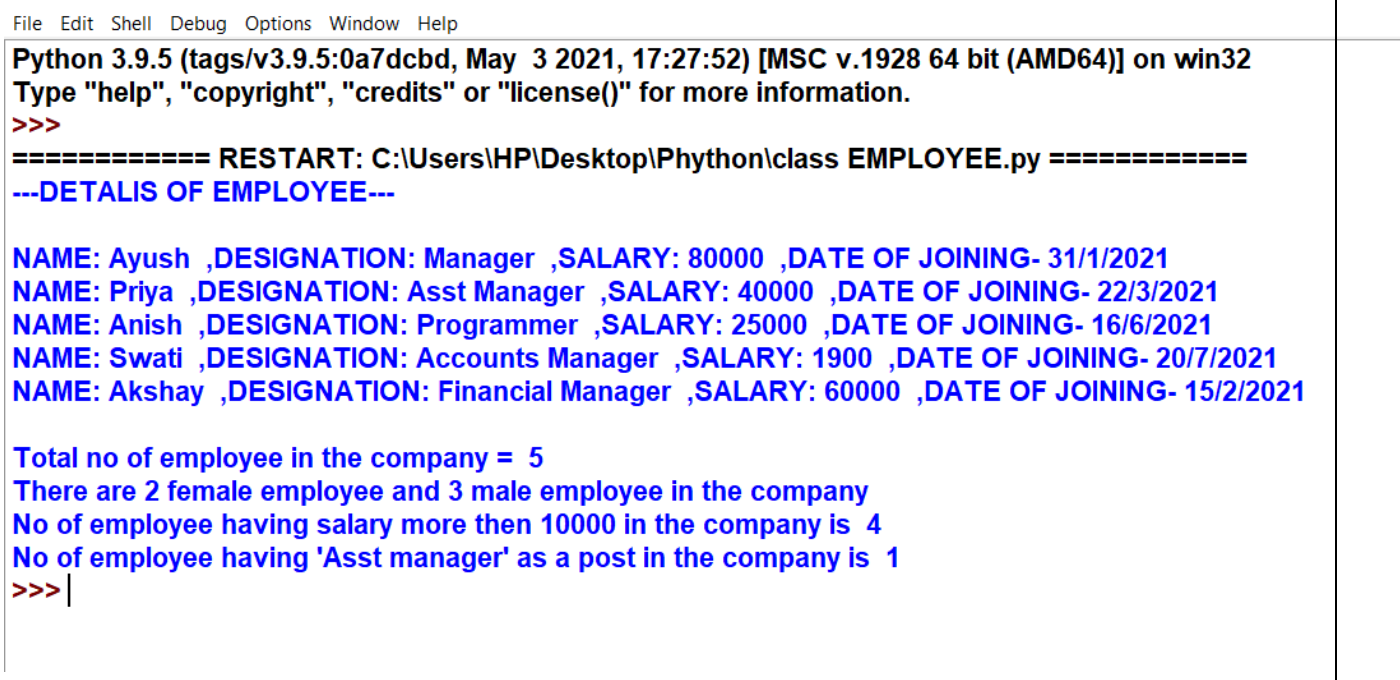
Problem Statement: Write a Program in Python Create class EMPLOYEE for storing details (Name, Designation, gender, Date of Joining and Salary). Define function members to compute a)total number of employees in an organization b) count of male and female employee c) Employee with salary more than 10,000 d) Employee with designation “Asst Manager”

Program Screenshot/Program

File Edit Format Run Options Window Help

```
class EMPLOYEE():
    count = 0
    fcount = 0
    mcount = 0
    sal_count = 0
    desig_count = 0

    def __init__(self,name,desig,gender,date,salary):
        self.name=name
        self.desig=desig
        self.gender=gender
        self.date=date
        self.salary=salary
        EMPLOYEE.count += 1
        if (self.gender == "f"):
            EMPLOYEE.fcount += 1
        else:
            EMPLOYEE.mcount += 1

        if (self.salary > 10000):
            EMPLOYEE.sal_count += 1
        if (self.desig == "Asst Manager"):
            EMPLOYEE.desig_count += 1

    def display_count(self):
        print("\nTotal no of employee in the company = ",EMPLOYEE.count)

    def count_gender(self):
        print("There are %d female employee and %i male employee in the company" % (EMPLOYEE.fcount,EMPLOYEE.mcount))


    def count_sal(self):
        print("No of employee having salary more then 10000 in the company is ",EMPLOYEE.sal_count)

    def count_desig(self):
        print("No of employee having 'Asst manager' as a post in the company is ",EMPLOYEE.desig_count)

    def displayDetails(self):
        print("NAME:",self.name," ,DESIGNATION:",self.desig," ,SALARY:",self.salary," ,DATE OF JOINING-",self.date)

print("---DETALIS OF EMPLOYEE---")
Emp1= EMPLOYEE("Ayush","Manager","m","31/1/2021",80000)
Emp2= EMPLOYEE("Priya","Asst Manager","f","22/3/2021",40000)
Emp3= EMPLOYEE("Anish","Programmer","m","16/6/2021", 25000)
Emp4= EMPLOYEE("Swati","Accounts Manager","f","20/7/2021",1900)
Emp5= EMPLOYEE("Akshay","Financial Manager","m","15/2/2021",60000)
print()

Emp1.displayDetails()
Emp2.displayDetails()
Emp3.displayDetails()
Emp4.displayDetails()
Emp5.displayDetails()

Emp1.display_count()
Emp1.count_gender()
Emp1.count_sal()
Emp1.count_desig()
```

Output Screenshot:



Conclusion :

In this assignment we have studied and created classes and objects and successfully performed operations using it.

***