Tab 1

SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

(UNIVERSITY ESTABLISHED UNDER SECTION 3 OF UGC ACT 1956)
ENATHUR, KANCHIPURAM – 631 561

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this is the bonafide record of work done by **Mr/Ms.** _____,

with **Reg. No** _____of **II-B.E.(CSE)** during the academic year 2025 – 2026.

Station:

Date:

Staff-in-charge **Head of the Department**

Submitted for the Practical examination held on_____.

SL.NO	Program Name	Date	Page no	Signature

PYTHON RECORD S.RESHMITHA 11249A339

1] name=(input("enter your name"))
print("Hello",name)

OUTPUT:

enter your name Reshmitha
Hello Reshmitha

2] A=10
   B=20
   Print("The total is",A+B)

OUTPUT:

The Total is 30

3] Ask the user three numbers Add together the first two numbers and then multiply .This total by the third Display the answer

  A=10
B=20
C=30
print("The answer is",A + B * C)

OUTPUT:

The answer is 610

4]Ask the user the total price of the bill then ask how many diners there divide the total bill by the no of

```
total_price = int(input("Enter total price:
"))
diners = int(input("Enter number of diners:
"))
print("Total price:", total_price)
print("Diners are:", diners)
print("Each person must pay:", total_price
/ diners)
```

OUTPUT:

```
Enter total price: 350
Enter number of diners: 5
Total price: 350
Diners are: 5
Each person must pay: 70.0
```

5]Number of Days,Hours,Minutes and Seconds

```
a = int(input("Enter number of days: "))
print("Number of hours:", a * 24)
print("Number of minutes:", a * 24 * 60)
print("Number of seconds:", a * 24 * 60 *
60)
```

OUTPUT:

```
Enter number of days: 2
Number of hours: 48
Number of minutes: 2880
Number of seconds: 172800
```

6] Find the area (A = ½ × h × (a + b))

```
h = float(input("Enter height: "))
a = float(input("Enter base a: "))
b = float(input("Enter base b: "))
A = 0.5 * h * (a + b)
print("Area =", A)
```

Enter temperature in Celsius: 25
Temperature in Fahrenheit: 77.0
Enter temperature in Fahrenheit: 77
Temperature in Celsius: 25.0

OUTPUT:

```
Enter height: 10
Enter base a: 8
Enter base b: 6
Area = 70.0
```

7]Convert Celsius ↔ Fahrenheit

```
# Celsius to Fahrenheit
c = float(input("Enter temperature in
Celsius: "))
f = (9/5 * c) + 32
print("Temperature in Fahrenheit:", f)
```

```
# Fahrenheit to Celsius
f = float(input("Enter temperature in
Fahrenheit: "))
c = (f - 32) * 5/9
print("Temperature in Celsius:", c)
```

OUTPUT:

```
Enter temperature in Celsius: 25
Temperature in Fahrenheit: 77.0
Enter temperature in Fahrenheit: 77
Temperature in Celsius: 25.0
```

8] Simple Calculator

```
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))

print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Modulus:", a % b)
```

OUTPUT:

```
Enter first number: 15
Enter second number: 4
Addition: 19.0
Subtraction: 11.0
```

```
Multiplication: 60.0
Division: 3.75
Modulus: 3.0
```

9] State the output of the code

```
num1 = '10'
num2 = '20'
sum = num1 + num2
print(sum)
```

OUTPUT:

```
1020
```

10]Identify the error

```
num1 = '10'
num2 = 20.65
sum = num1 + num2
print(sum)
```

OUTPUT:

Error: TypeError — you cannot add a string (`'10'`) and a float (`20.65`).

Fix:

```
sum = float(num1) + num2
```

```
print(sum)
```

11] Invoice Calculation

```
cost = 5000
cgst = cost * 9 / 100
sgst = cost * 9 / 100
total = cost + cgst + sgst

print("Cost of Production:", cost)
print("Add: CGST @9% =", cgst)
print("Add: SGST @9% =", sgst)
print("Total Cost of Product:", total)
```

OUTPUT:

```
Cost of Production: 5000
Add: CGST @9% = 450.0
Add: SGST @9% = 450.0
Total Cost of Product: 5900.0
```

Operators

12] Convert grams to kilograms and grams

```
grams = int(input("Enter weight in grams:
"))
kg = grams // 1000
```

```
remaining_grams = grams % 1000

print("Weight:", kg, "kg and",
remaining_grams, "grams")
```

OUTPUT:

```
Enter weight in grams: 2560
Weight: 2 kg and 560 grams
```

13] Identity Operators

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]

print(x1 is y1)
print(x2 is y2)
print(x3 is y3)
```

OUTPUT:

```
True
```

```
True
False
```

Conditional Statements

1. Student Mark Analysis

```
mark = int(input("Enter mark: "))

if mark >= 90:
    print("Grade: A+")
elif mark >= 75:
    print("Grade: A")
elif mark >= 60:
    print("Grade: B")
elif mark >= 50:
    print("Grade: C")
else:
    print("Fail")
```

Output:

```
Enter mark: 78
Grade: A
```

2. Number Check (type, positive/negative, even/odd)

```
data = input("Enter something: ")

    if num > 0:
        print("Positive number")
        if num % 2 == 0:
            print("Even")
        else:
            print("Odd")
    elif num < 0:
        print("Negative number")
    else:
        print("Zero")
else:
    print("Not a number. Data type:",
type(data).__name__)
```

Output Example 1:

```
Enter something: 12
Positive number
Even
```

Output Example 2:

```
Enter something: hello
Not a number. Data type: str
```

3. Simple Interest (SI)

```
principle = float(input("Enter principle amount: "))
rate = 5 if principle > 100000 else 3.5
years = float(input("Enter number of years: "))

SI = (principle * rate * years) / 100
print("Simple Interest =", SI)
```

Output:

```
Enter principle amount: 150000
Enter number of years: 3
Simple Interest = 22500.0
```

looping Statements

1. Sum of 10 Consecutive Numbers (1 to 10)

```
i = 1
total = 0

while i <= 10:
    total += i
    i += 1
```

```
print("Sum of 10 consecutive numbers =", total)
```

Output:

```
Sum of 10 consecutive numbers = 55
```

2. Sum of Digits of a Number

```
num = int(input("Enter a number: "))
sum_digits = 0

while num > 0:
    sum_digits += num % 10
    num //= 10

print("Sum of digits =", sum_digits)
```

Output:

```
Enter a number: 12345
Sum of digits = 15
```

RANGE FUNCTION

```
Sum=0
print(“Numbers from 1 to 20 which are not divisible by 2,3,or 5”)
for i in range(1,20):
        if i%2==0 or i%3==0 or i%5==0:
                print(“”)
        else:
                print(i)
        sum=sum+i
```

```
print("Sum of Even numbers from 1 to 10 is = ",sum)
```

Output

```
Numbers from 1 to 20 which are not divisible by 2, 3, and 5
1
7
11
13
17
19
Sum of Even numbers from 1 to 10 is = 68
```

2 Write a program to print Fibonacci series up to 8.

```
First_Number = 0
Second_Number = 1
Fibonacii Series = 0 1 1 2 3 5 8 13 21 34 55
```

```
First_Number=int(input("Please enter First Number:"))
Second_Number=int(input("Please enter First Number:"))
Limit=int(input(" Number of Fibonacci Numbers to be Print: "))
print(First_Number,end=" ")
print(Second_Number,end=" ")
for i in range(Limit+1):
        sum=First_Number+Second_Number
        First_Number=Second_Number
        Second_Number=sum
        print(sum,end=" ")
```

Output

```
Please enter First Number:0
Please enter First Number:1
Number of Fibonacci Numbers to be Print: 8
0 1 1 2 3 5 8 13 21 34 55
Nested loop
3 .for i in range(1,4,1): #Outer Loop
        for j in range(1,4,1): #Inner Loop
```

```
                print("i = ",i," j = ",j," i + j =",i + j)
print("End of Program")
```

Output

```
i = 1 j = 1 i + j = 2
i = 1 j = 2 i + j = 3
i = 1 j = 3 i + j = 4
i = 2 j = 1 i + j = 3
i = 2 j = 2 i + j = 4
i = 2 j = 3 i + j = 5
i = 3 j = 1 i + j = 4
i = 3 j = 2 i + j = 5
i = 3 j = 3 i + j = 6
```

4 Write a program to display the pattern of stars given as follows:

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

```
print(" Star Pattern Display")
num=7
x=num
for i in range(1,6,1):
        num=num-1;
                for j in range(1,num,1):
                        print(" * ",end=" ")
                        x=num-1
                print()
print("End of Program")
```

Output

```
Star Pattern Display
* * * * *
* * * *
```

* * *
* *
*
End of Program

5 Write a program to display the pattern of stars given as follows:

```
*
* *
* * *
* * * *
* * * * *
print(“ Star Pattern Display”)
num=1
x=num
for i in range(1,6,1):
num=num+1;
for j in range(1,num,1):
print(“ * “,end=” “)
x=num+1
print()
print(“End of Program”)
```

Output

```
Star Pattern Display
*
* *
* * *
* * * *
* * * * *
End of Program
```

6 Write a program to display the pattern of numbers given as follows:

```
1
1 2
1 2 3
```

1 2 3 4
1 2 3 4 5

```
print(“ Number Pattern Display”)
num=1
x=num
for i in range(1,6,1):
num=num+1;
for j in range(1,num,1):
print(j, end=” “)
x=num+1
print()
print(“End of Program”)
```

Output
Number Pattern Display
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
End of Program

7 Write a program to print the numbers from one to fi ve using the while loop.

```
count=0 #initialize the counter
while count<=5: # Test condition
        print(“Count = “,count) # print the value of count
        count=count+1 # Increment the value of count by 1
```

Output
Count = 0
Count = 1
Count = 2
Count = 3
Count = 4
Count = 5

8 Write a program to find the sum of the digits of a given number.
For example, if a user enters 123. The program should return (3+2+1), i.e. 6 as the sum of all the digits in a number.

```
num=int(input("Please Enter the number:"))#Read Number from User
x=num #Assign value of num to x
sum=0
rem=0
while num>0:
rem=num % 10
num=num // 10
sum=sum + rem
print("Sum of the digits of an entered number ",x," is = ",sum)
```

Output
Please Enter the number: 12345
Sum of the digits of an entered number 12345 is = 15

9 Write a program to display the reverse of the number entered.

For example, if a user enters 12345. The program should return (54321), i.e. the reverse of the number entered.

```
num =int(input("Please Enter the number: "))
x=num
rev=0
while num>0:
rem=num % 10
num=num // 10
rev=rev*10+rem
print("Reverse of a entered number ",x," is = ",rev)
```

Output
Please Enter the number: 8759
Reverse of a entered number 8759 is = 9578

10 Write a program to print the sum of the numbers from 1 to 20 (1 and 20 are included) that are divisible by 5 using the while loop.

```
count=1
sum=0
while count<=20:
if count%5 == 0:
sum=sum+count
count=count+1
print(“The Sum of Numbers from 1 to 20 divisible by 5 is: “,sum)
```

Output

The Sum of Numbers from 1 to 20 divisible by 5 is: 50

11 Write a program using the while loop to print the factorial of a number.

Factorial of 6 = 6*5*4*3*2*1 = 720

```
Num=int(input(“Enter the number:”))
fact=1
ans=1
while fact<=num:
ans=ans*fact
fact=fact+1
print(“Factorial of”,num,” is: “,ans)
```

Output

Enter the number:6

Factorial of 6 is: 720

Budget Allocation by Ratio

```
total_budget = float(input("Enter total budget: "))
marketing_ratio = float(input("Enter marketing ratio:
"))
development_ratio = float(input("Enter development
ratio: "))
```

```
operations_ratio = float(input("Enter operations ratio: "))

total_ratio = marketing_ratio + development_ratio + operations_ratio

marketing_budget = total_budget * (marketing_ratio / total_ratio)
development_budget = total_budget * (development_ratio / total_ratio)
operations_budget = total_budget * (operations_ratio / total_ratio)

print(f"Marketing budget: {marketing_budget:.2f}")
print(f"Development budget: {development_budget:.2f}")
print(f"Operations budget: {operations_budget:.2f}")
```

Input:

```
Enter total budget: 10000
Enter marketing ratio: 3
Enter development ratio: 5
Enter operations ratio: 2
```

Output:

```
Marketing budget: 3000.00
Development budget: 5000.00
Operations budget: 2000.00
```

---

2 Social Media Reach Estimation

Code:

```
followers = int(input("Enter followers: "))
followers_per_person = int(input("Enter followers per
person: "))
growth_percent = float(input("Enter growth percentage:
"))

estimated_reach = followers * followers_per_person * (1
+ growth_percent / 100)
print("Total Estimated Reach =", int(estimated_reach))
```

🧾 **Input:**

```
Enter followers: 500
Enter followers per person: 200
Enter growth percentage: 10
```

Output:

```
Total Estimated Reach = 10000
```

Total Stars Counted

Code:

```
nights = int(input("Enter number of nights: "))
total_stars = 0

for i in range(nights):
    stars = int(input(f"Enter stars counted on night {i+1}: "))
    total_stars += stars
```

```
print("Total stars counted:", total_stars)
```

Input:

```
Enter number of nights: 4
Enter stars counted on night 1: 20
Enter stars counted on night 2: 35
Enter stars counted on night 3: 25
Enter stars counted on night 4: 40
```

Output:

```
Total stars counted: 120
```

4 Rectangle Pattern Printing

Code:

```
height = int(input("Enter height: "))
width = int(input("Enter width: "))

for i in range(height):
    print("#" * width)
```

Input:

```
Enter height: 5
Enter width: 3
```

Output:

```
###
###
###
###
```

###

. Palindrome Check + Digit Occurrence Count (Python Program)

Program

```
num = input("Enter a number: ")

# Check palindrome
if num == num[::-1]:
    print("It is a palindrome.")
else:
    print("It is not a palindrome.")

# Count digit occurrences
count = {}
for d in num:
    if d.isdigit():      # ensure only digits
        count[d] = count.get(d, 0) + 1

print("Digit occurrences:")
for digit, c in count.items():
    print(f"{digit} → {c} time(s)")
```

2. Your vowel-count program has an error

What is the error?

From the output:

```
Letter 5:
Number of vowels in the puzzle: 4
```

You entered **only 4 letters**, because for Letter 5 you pressed **Enter without typing anything.**

So the program counts:

- a → vowel
- e → vowel
- y → **not vowel**
- i → vowel
- "" (empty string) → not vowel

Thus total vowels = **3, not 4.**

But your program showed **4** → means the program incorrectly counted something.

Reason:
`ch = input()` may take an empty string, but your loop still appends it and checks.
Your code is correct — **your sample output is wrong.**

But to make the program robust, we can fix the case where the user presses Enter without typing anything.

Corrected Vowel Counting Program

Improved Program (Rejects empty input)

```
# Program to count vowels in given letters
n = int(input("Number of letters: "))
letters = []
for i in range(n):
    ch = input(f"Letter {i+1}: ").strip().lower()
```

```
    # Force the user to enter exactly one letter
    while len(ch) != 1 or not ch.isalpha():
        print("Invalid input! Please enter one letter only.")
        ch = input(f"Letter {i+1}: ").strip().lower()
    letters.append(ch)
# Define vowels
vowels = "aeiou"
# Count vowels
count = 0
for ch in letters:
    if ch in vowels:
        count += 1
print("Number of vowels in the puzzle:", count)
```

List operations and methods :

```
my_list = ['p','r','o','g','r','a','m','m','e']
print(my_list[2:5])
print(my_list[5:])
print(my_list[:])
list1 = [10, 15, 11, 65, 30]
list1.insert(1,80)
print('INSERTED ELEMENT IN THE LISTS :', list1)

list1.sort()
print('SORTED ELEMENT IN THE LIST IS :', list1)
        list1.reverse()
```

```
print('REVERSED ELEMENT IN THE LIST IS :',list1.pop(3,)
print('REMOVED ELEMENT IN THE LIST IS :', list1)
print('LARGEST ELEMENT IN THE LIST IS :',max(list1))
print('SMALLEST ELEMENT IN THE LIST IS :', min(list1))
```

Output

```
['o', 'g', 'r']
['a', 'm', 'm', 'e']
['p', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'e']
INSERTED ELEMENT IN THE LIST IS : [10,
80, 15, 11, 65, 30] SORTED ELEMENT IN
THE LIST IS : [10, 11, 15, 30, 65, 80]
REVERSED ELEMENT IN THE LIST IS :
[80, 65, 30, 15, 11, 10] REMOVED ELEMENT
IN THE LIST IS : [80, 65, 30, 11, 10]
LARGEST ELEMENT IN THE LIST IS : 80
SMALLEST ELEMENT IN THE LIST IS : 10
```

List Comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

for x in fruits:
  if "a" in x:
    newlist.append(x)

print(newlist)
```

OUTPUT:

```
['apple', 'banana', 'mango']
```

List Program :

```
prices = [1.09, 23.56, 57.84, 4.56, 6.78]
TAX_RATE = .08
def get_price_with_tax(price):
                return price * (1 + TAX_RATE)



final_prices = map(get_price_with_tax, prices)
print( final_prices)
l = [13, 4, 5, 3, 4]
```

OUTPUT

[1.1772000000000002, 25.4448, 62.467200000000005, 4.9248, 7.3224000000000001]

Display the frequency of each element in list

```
l=[13,4,5,3,4]
l1=[]  # 2,2 , 1
l2=[]  # 3  4   5
print("Element \t \t frequency")
for i in l:
    if i not in l2:
        x=l.count(i)  # i= 3
        l1.append(x)
        l2.append(i)
for i in range (len(l1)):
    print(l2[i],"\t \t \t " , l1[i])
```

OUTPUT

l = [13, 4, 5, 3, 4]

Element	frequency
13	1
4	2
5	1
3	1

Tuples Operations and methods

PROGRAM:

```
tuple1 = (1, 'a', 2, 'b')
tuple2 = (2, 'c', 4, 'd')
print(type(tuple1))
print(len(tuple1))
print(tuple1 + tuple2)
print(tuple2 * 3)
```

OUTPUT

```
<class 'tuple'>
4
(1, 'a', 2, 'b', 2, 'c', 4, 'd')
(2, 'c', 4, 'd', 2, 'c', 4, 'd', 2, 'c', 4, 'd')
```

```
t=tuple()
n=int(input("How many values you want to enter: "))
for i in range(n):
    a=input("Enter Number: ")
    t=t+(a,)
print("Entered Numbers are: ")
print(t)
```

OUTPUT

```
Entered Numbers are:
```

('10', '20', '30')

c) Demonstrate on create and print Dictionaries

```
country_capitals =
{
"United States": "Washington D.C.", "England": "London", "Germany":
Berlin", "Canada": "Ottawa "  }

print(len(country_capitals))
print(country_capitals)
country_capitals["Italy"] = "Rome"
print(country_capitals)
del country_capitals["Germany"]
print(country_capitals)
country_capitals.clear()
print(country_capitals)
```

Output:

```
Length of Dictonary: 4
{'United States': 'Washington D.C.',
'England': 'London', 'Germany':
'Berlin', 'Canada': 'Ottawa '}
```

```
{'United States': 'Washington D.C.',
'England': 'London', 'Germany':
'Berlin', 'Canada': 'Ottawa ', 'Italy':
'Rome'}
{'United States': 'Washington D.C.',
'England': 'London', 'Canada': 'Ottawa ',
'Italy': 'Rome'}
{}
```

String operations and methods:

```
a = "Hello, World! "
print(a[2:5])
print(a[:5])
print(a[2:])
print(a[-5:-2])
print(a.upper())
print(a.lower())
print(a.strip())   # [‘H’, ‘e’,’l’,]
print(a.replace("H", "J"))
print(a.split(","))
print(len(a))
```

OUTPUT:

```
llo
Hello
llo, World!
rld
HELLO, WORLD!
hello, world!
Hello, World!
Jello, World!
['Hello', ' World! ']
14
```

```
a="Hello"
b="World"
c=b+a
print(c) # WorldHello
c=b+"\t "+a # World Hello


print(c)
```

OUTPUT:

```
WorldHello
World	Hello
```

```
str1 = "Hello"
str2 = " Hello"
print(str1 is str2 )
print(str1 is not  str2 )
print(str1*3) # prints HelloHelloHello
print(str1+str2) # prints Hello world
print(str1[4]) # prints o
print(str1[2:4]) # prints ll
print('w' in str1) # prints False as w is not present in str1
print('wo' not in str2) # prints false as Wo is present in str2.
print(r'Hello\n world') # prints Hello\n world as it is  written
print("The string str1 : %s"%(str1)) # prints The string str :
Hello
```

OUTPUT:

```
False
True
HelloHelloHello
Hello Hello
o
ll
False
True
```

Hello\n world

The string str1 : Hello

Python Example Program - split() in Python

Example:

```
str1 = "Java is a programming language"
str2 = str1.split()
# Displaying result
print(str1)
print(str2)

str2 = str1.split(sep=',')
# Displaying result
print(str1)
print(str2)
```

OUTPUT:

Java is a programming language

['Java', 'is', 'a', 'programming', 'language']

Java is a programming language

['Java is a programming language']

Python Example Program - find() in Python

Example:

```
str1 = "python is a programming language"
str2 = str1.find("is")
str3 = str1.find("java")
str4 = str1.find("n",5)
str5 = str1.find("i",5,25)
print(str2,str3,str4,str5)


sentence = (
    "The rocket, who was named Ted, came back "
    "from Mars because he missed his friends."  )
 def is_consonant(letter):
    vowels = "aeiou"
    return letter.isalpha() and letter.lower() not in vowels
print( [char for char in sentence if is_consonant(char)])
```

Output :

```
['T', 'h', 'r', 'c', 'k', 't', 'w', 'h', 'w', 's', 'n', 'm', 'd',
 'T', 'd', 'c', 'm', 'b', 'c', 'k', 'f', 'r', 'm', 'M', 'r', 's', 'b',
 'c', 's', 'h', 'm', 's', 's', 'd', 'h', 's', 'f', 'r', 'n', 'd', 's']
```

Randomly choose either heads or tails (“h” or “t”). Ask the user to make their choice. If their choice is the same as the randomly selected value, display the message “You win”, otherwise display “Bad luck”. At the end, tell the user if the computer selected heads or tails.

```
import random
value = random.choice(["h", "t"])
choice = input("Make your choice!!(h or t) ").lower()   //h
if value == choice: // t==h
    print("You win")
else:
    print("Bad luck")
if value == "h":
    print("The computer selected heads")
elif value =="t":
    print("The computer selected tails")
OUTPUT
Make your choice!!(h or t) h
Bad luck
The computer selected tails
```

```
import random

colours = ["black", "white", "blue", "red", "yellow"]
messages = {
    "black": "You are such a BLACK luck guy.",
    "white": "Too naive to choose WHITE.",
    "blue": "You are probably feeling BLUE right now.",
    "red": "Is your face turning RED right now?",
    "yellow": "I bet your future is as bright as YELLOW."
}
```

```
colour = random.choice(colours)

while True:
    guess = input("Pick one colour: black, white, blue, red, yellow
").lower()

    if guess == colour:
        print("Well done! YOU WON THE GAME ")
        break
    else:
        print(messages[colour])
```

OUTPUT

Pick one colour: black, white, blue, red, yellow black

You are probably feeling BLUE right now.

Pick one colour: black, white, blue, red, yellow red

You are probably feeling BLUE right now.

Pick one colour: black, white, blue, red, yellow blue

Well done! YOU WON THE GAME

```
from collections import Counter
```

```
counts = Counter(['a', 'b', 'a', 'c', 'b']) # Output: Counter({'a': 2, 'b': 2, 'c': 1})

print(counts)
from collections import deque
d = deque([1, 2, 3])
print(d)
d.appendleft(0) # deque([0, 1, 2, 3])
print(d)
d.pop() # 3, d is now deque([0, 1, 2])
print(d)
```

OUTPUT:

deque([1, 2, 3])

deque([0, 1, 2, 3])

deque([0, 1, 2])

`namedtuple`

```
from collections import namedtuple

Point = namedtuple('Point', ['x', 'y'])

p = Point(10, 20)

print(p.x) # 10
```

`defaultdict`:

```
from collections import defaultdict

s = [('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]

d = defaultdict(list)

for k, v in s:

    d[k].append(v)
```

```
print(d)
```

OUTPUT

defaultdict(<class 'list'>, {'yellow': [1, 3], 'blue': [2, 4], 'red': [1]})

FILE HANDLING

1. Write a program to write the sentences given below the fi le Demo1.txt.

Hello, How are You?
Welcome to The chapter File Handling.
Enjoy the session.

```
def main():
        obj1 = open("Demo1.txt","w") #Opens file in Write mode
        obj1.write(" Hello, How are You ? \n")
        obj1.write(" Welcome to The chapter File Handling. \n ")
        obj1.write(" Enjoy the session. \n ")
        obj1.close()
main() # Call to main function
```

OUTPUT

Hello, How are You?

Welcome to The chapter File Handling.

Enjoy the session.

Write numbers from 1 to 20 to the output fi le WriteNumbers.txt.

```
def main():
        obj1 = open("WriteNumbers.txt","w") #Open File in Write mode
        for x in range(1,21): # Iterates from 1 to 20
                x=str(x) # Convert Number to String
                obj1.write(x) # Write Number to a output file
                obj1.write(" ") # Space to separate Numbers
        obj1.close() # Close File
main() # Call to main function
```

OUTPUT

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Generate 50 random numbers within a range 500 to 1000 and write them to fi le

WriteNumRandom.txt.

```
from random import randint # Import Random Module
fp1 = open("WriteNumRandom.txt","w") # Open file in write mode
        for x in range(51): #Iterates for 50 times
                x = randint(500,1000) #Generate one random number
                x = str(x) #Convert Number to String
                fp1.write(x + " ") #Write Number to Output file
fp1.close() #Finish Writing Close the file
```

OUTPUT:

743 612 998 523 875 690 ... (up to 50 numbers)

Write a program to add the content of a fi le numbers.txt and display the sum of all the numbers present in the fi le.

```
fp1 = open("numbers.txt","r")
num = int(fp1.readline())
print(num)
sum = 0
print('The ', num ,' numbers present in the file are as follows:')
for i in range(num):
        num1 = int(fp1.readline())
        print(num1)
        sum = sum + num1
print('Sum of all the numbers (except first):')
print(sum)
Output
5
The 5 numbers present in the file are as follows:
2
4
```

File Handling 365

6
8
10
Sum of all the numbers (except first):
30

Write a program to accept string/sentences from the user till the user enters “END” to. Save the data in a text file and then display only those sentences which begin with an uppercase alphabet.

Answer

```
f = open("new.txt", "w")
while True:
    st = input("Enter next line:")
    if st == "END":
        break
    f.write(st + '\n')
f.close()

f = open("new.txt", "r")
while True:
    st = f.readline()
    if not st:
break
    if st[0].isupper():
        print(st)
f.close()
```

Output

```
Enter next line:Hello world
Enter next line:welcome to
Enter next line:Python programming
Enter next line:END
Hello world

Python programming
```

```
import os.path

import sys

fname = input("Enter the filename : ")

if not os.path.isfile(fname):

 print("File", fname, "doesn't exists")

 sys.exit(0)

infile = open(fname, "r")

lineList = infile.readlines()

lineCount = int(input("Enter the number of lines you want to display :
"))

for i in range(lineCount):

    print(i+1, ":", lineList[i], end="")

    word = input ("Enter a word : ")

    cnt = 0

for line in lineList:

 cnt += line.count(word)

    print("The word", word, "appears", cnt, "times in the file")
```

OUTPUT

```
Enter the filename : Demo1.txt

Enter the number of lines you want to display : 2

1 : Hello, How are You?

2 : Welcome to The chapter File Handling.

Enter a word : You?
```

The word You? appears 1 times.

Wel.txt

Welcome to cse

```
f = open("Wel.txt", "r")

print(f.read(3))
```

Output :

Wel

```
f = open("Wel.txt", "r")

print(f.read(-1))# complete file contents will be displayed
```

Output :

Welcome to cse

```
f = open("Wel.txt", "r")

print(f.readable())

f = open("Wel.txt", "r")

f.seek(2)

print(f.readline())

OUTPUT:

lcome to cse
```

Ask the user to enter 1, 2 or 3. If they select anything other than 1, 2 or 3 it should display a suitable error message. If they select 1, ask the user to enter a school subject and save it to a new file called “Subject.txt”. It should overwrite any existing file with a new file. If they select 2, display the contents of the “Subject.txt” file. If they select 3, ask the user to enter a new subject and save it to the file and then display the entire contents of the file. Run the program several times to test the options.

```
count = 0

while count < 5:

    print('1) Create a new file')

    print('2) Display the file')

    print('3) Add a new item to the file ')

    selection_input = int(input('Make a selection 1, 2 or 3 : '))

    if selection_input == 1:

        file = open('Subject.txt', 'w')

        subject_name = str(input('Enter school subject name : '))

        file.write(subject_name + '\n')

        file.close()

    elif selection_input == 2:

        file = open('Subject.txt', 'r')

        print(file.read())
```

```
        file.close()
    elif selection_input == 3:
        file = open('Subject.txt', 'a')
        subject_name = str(input('Enter school subject name to add into file : '))
        file.write(subject_name + '\n')
        file.close()
    count = count + 1


file = open('Subject.txt', 'r')
print(file.read())
file.close()
```

OUTPUT

1) Create a new file

2) Display the file

3) Add a new item to the file

Make a selection 1, 2 or 3: 1

Enter school subject name: Mathematics

File created successfully!

EXCEPTION HANDLING

```
try:
  n = 0
  res = 100 / n
except ZeroDivisionError:
  print("You can't divide by zero!")
  n = int(input('enter value '))#GIVE INPUT AS a then check the program
  res = 100 / n
  print(res)
except ValueError:
  print("Enter a valid number!")

else:
  print("Result is", res)

finally:
  print("Execution complete.")
```

```
You can't divide by zero!

enter value a

Traceback (most recent call last):

  File "program.py", line 5, in <module>

    n = int(input('enter value '))

ValueError: invalid literal for int() with base 10: 'a'

Execution complete.
```

Catching Multiple Exceptions

```
a = ["10", "twenty", 30]  # Mixed list of integers and strings
try:
```

```
    total = int(a[0]) + int(a[1])  # 'twenty' cannot be converted to int

except (ValueError, TypeError) as e:
    print("Error", e)

except IndexError:
    print("Index out of range.")
OUTPUT
```

Error invalid literal for int() with base 10: 'twenty'

```
n = 10
res = n / 0

try:
    numerator = int(input("Enter a numerator: "))
   denominator = int(input("Enter a denominator: "))
   result = numerator / denominator
except ValueError:
   # Handles invalid input (e.g., non-integer)
   print("Invalid input. Please enter an integer.")
except ZeroDivisionError:
   # Handles division by zero
OUTPUT
```

Enter a numerator: 10

Enter a denominator: 2

Result is: 5.0

Enter a numerator: a

Invalid input. Please enter an integer.

```
def divide_num(num_list):
  for num in num_list:
```

```
    print(10/num)

num_list = [5, 6, 0, 7]
divide_num(num_list)
```

```
Output :
2.0
1.66666666666666
Traceback (most recent call last):

  File "F:/NETJS/NetJS_2017/Python/Test/Test.py", line 6, in <module>
    divide_num(num_list)
  File "F:/NETJS/NetJS_2017/Python/Test/Test.py", line 3, in divide_num
    print(10/num)
ZeroDivisionError: division by zero
```

```
def divide_num(num):
  try:
    f = open("testfile", "a")
    r = 10/num
    f.write("Result 10/%d is %d" %(num,r))
  except ZeroDivisionError as error:
    print(error)
    print('Zero is not a valid argument here')
  except FileNotFoundError as error:
    print(error)
    print('Passed file does not exist')
  finally:
    print('closing file')

    f.close()
```

OUTPUT

```
divide_num(0)
```

```
division by zero
```

Zero is not a valid argument here

SHELVE

import shelve

1. Open a shelf file

The 'my_data.db' argument specifies the filename.

'c' mode creates the file if it doesn't exist, and opens it for read/write.

Using 'with' ensures the shelf is properly closed even if errors occur.

with shelve.open('my_data.db', 'c') as shelf:

2. Store variables (like dictionary key-value pairs)

shelf['name'] = 'Alice'

shelf['age'] = 30

shelf['favorite_numbers'] = [7, 13, 42]

shelf['is_student'] = False

print("Data stored in 'my_data.db'.")

3. Retrieve variables from the shelf file

with shelve.open('my_data.db', 'r') as shelf:

Access stored values using their keys

retrieved_name = shelf['name']

retrieved_age = shelf['age']

retrieved_numbers = shelf['favorite_numbers']

retrieved_student_status = shelf['is_student']

print(f"\nRetrieved Name: {retrieved_name}")

print(f"Retrieved Age: {retrieved_age}")

print(f"Retrieved Favorite Numbers: {retrieved_numbers}")

print(f"Retrieved Student Status: {retrieved_student_status}")

4. Update a variable in the shelf

with shelve.open('my_data.db', 'w') as shelf: # 'w' mode for write access

shelf['age'] = 31

print("\nAge updated to 31.")

5. Verify the update

with shelve.open('my_data.db', 'r') as shelf:

print(f"Updated Age: {shelf['age']}")

6. Delete a variable from the shelf

with shelve.open('my_data.db', 'w') as shelf:

del shelf['is_student']

print("\n'is_student' variable deleted.")

7. Check if the variable exists (this will raise a KeyError if not found)

try:

with shelve.open('my_data.db', 'r') as shelf:

print(f"Is Student (after deletion): {shelf['is_student']}")

except KeyError:

print("'is_student' variable not found after deletion.")

OUTPUT

'is_student' variable deleted.

'is_student' variable not found after deletion.

Tab 2