Date:-

Task-NO:9 :-

Implement exceptions and exceptional handling in python

Aim :- To implement Exceptions and exceptional handling in python.

problem-9.1 :- You are developing a python processes a list of students grades. The program is designed to allow the user to select a grade by aspecifying an index number. However, you need to ensure that the program handles takes where the user inputs an index that is out of range, i.e., an indesc does not exist in the disk.

Algorithm :-

1) Start the program.
2) Initialize a list of grades (e.g., [85,90,78,92,88)].
3) Prompts the user to enter the index of the grade obey wish to view.
4) Attempts to display the grade at the specified index.
5) If the indesc is out of range, catches the Indesc error and prints an error message. Invalid Indesc. please enter a valid index".

Program :-

```
# initialize the list of grades.
grades = [85,90,48,92,88]

# Display the grades list
print ("Grades list :", grades).
```

output:-

Grades List : [85, 90, 78, 92, 88].

Enter the index of the grade you want to view: 10.

Invalid index. please enter a valid index.

output

Enter the num 10

Enter the denominator : 0

Error:

Error : Division by zero is not allowed.

```
try:
    index = int (input ("Enter the index of the grade you
                        want to view:")).
    print (f"The grades as index {index} is: {grades [index]}")
except Index error:
    print ("Invalid Index. Please enter a valid index:").
except value error:
    print ("Invalid input. Please enter a numerical
                                                index).
```

Problem-9.2 :- you are developing a python calculator program that performs Basic arithmetic operations one of the key functionalities is to divide two numbers entered by the user. However, dividing by zero is not allowed and would cause program to crash if not handled properly.

Algorithm :-

-> Start the program.

-> prompts the user to enter the two numbers: a numerator and a denominator.

-> attempts to divide the numerator by the denomina-tor.

4) If the denominator is zero, catches the zero Division error and displays an error message. "Error: Division by zero is not allowed".

Program :-

```
# function to perform division.
def divide - numbers ():
    try:
    # prompt the users to enter numerator
```

Output :-

```
Enter the num 10
Enter the denominator: 0
  Error:
  Error: Division by zero is not allowed.
```

```
        numerator = float(input("enter the numerator:"))
        denominator = float(input("enter the denominator:"))
        result = numerator / denominator
        print(f"result : {result}")
    except ZeroDivisionError:
        print("Error : Division by zero is not allowed:")
    except ValueError:
        print("Error : please enter valid numbers:")
divide_numbers()
```

problem 9.3: You are building a python application to determine if a person is eligible to vote Based on their age According to the rules, only individuals who are 18 years or older are allowed to vote. To enforce this rule, you decide to create a custom exception called Invalid age exception, which will be raised whenever an age below 18 is entered.

Algorithm :-

-> Define the custom exception.
-> Prompt the user for input.
-> check if the age is below 18
-> Raise an exception if the condition is met
-> Handle the exception with a custom error message.

Output

Enter a number = 15

Exception occured : Invalid age.

Program:

```
class Invalid age exception (Exception):
    "Raised when the input value is
        less than 18".
        Pass
number = 18
try:
    input-num = int(input("enter a number:"))
    if input -num < number:
        raise Invalid age exception
    else:
        Print ("eligible to vote")
except Invalid age exception:
    Print( "exception occured. Invalid age").
```

VEL TECH	
EX NO.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGN WITH DATE	

Result:- Thus, the program for implement exceptions and exceptional handling is executed & verified successfully.