Task 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 program that processes a list of students 'grades.

Algorithm:-

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

Program:-

```
# Initialize the list of grades
grades = [85, 90, 78, 92, 88]

# Display the grades list
print("Grades List", grades)

# prompt the user to enter the index of the grade they want to view try:
index = int (input ("Enter the index of the grade you want to view:"))
    # Attempt to display the grade at the specified index
    print (f"The grade at index {index} is: {grades [index]}")
except Index Error:
    # Handle the case where the index is out of range
    Print ("Invalid index. please enter a valid index.")
except value Error:
    # Handle the case where the input is not an integer
    print ("Invalid input please enter a numerical index".)
```

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.

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.

Algorithm:-

1. start the program
2. prompts the user to enter two number: a numerator and a denominator.
3. Attempts to divide the numerator by the denominator.
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 user to enter the numberator
       numerator = float (input ("Enter the numberator:"))
       # prompt the user to enter the denominator
        denominator = float (input ("Enter the denominator:"))
       # Attempt to perform divison
          result = numerator / denominator
          print (f"Result: {result}")
       except zero Division Error:
          # Handle division by zero error
       print f("Error: Division by zero is not allowed.")
       except value Error:
          # Handle invalid input that is not a number
          print ("Error: please enter valid numbers")

# Call the function to execute the division operation
divide_numbers()
```

Output:-

```
Enter the numerator: 10
Enter the denominator : 0
ERROR!
Error : Division by zero is not allowed.
```

o/p

Output:-

Enter a number: 15
Exception occured : Invalid Age

Problem 9.3:- You are building a python application to determine if a person is eligible to vote based on their age.

Algorithm:-

1) Define the custom exception.
2) prompt the user for input.
3) check if the age is below 18.
4) Raise an exception if the condition is met.
5) Handle the exception with a custom error message.

Program:-

```
# define python user-define exceptions
class Invalid Exception (Exception):
    "Raised when the input value is less than 18"
    pass
    # you need to guess this number
    number = 18

try:
    Input_num = int(input("Enter a number:"))
    if input_num < number:
        raise Invalid Age Exception
    else:
        print("Eligible to Voice")
except Invaild Age Exception:
    print("Exception occured: Invaild Age")
```

Result:- Thus the program for implement Exceptions and Exceptional handling is executed and verified successfully.

VEL TECH	
E. No.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	
RECORD (5)	5
TOTAL (20)	
SIGN WITH DATE	17/9.