24/09/25 Task 9. Implement Expections and Exceptional handling in python

Aim: To implement Exceptions and Exceptional handling in Python.

Algorithm:

1. Start the Program
2. Initialize a list of grades
3. prompts the user to enter the index of the grade wish they to view
4. Attempts to display the grade at the specified index
5. If the index is out of range, catches the index Error and prints an error message.

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 grade they want to view:"))
   # Attempt to display the grade at the specified index
   Printf ("The grede at index {index} is: {grades[index]}")
except Index Error:
  # Handle the case where the index is out of range
   Print ("Invaild index please enter a valid index")
  except ValueError:
   # Handle the case where the input is not an integer
    Print ("Invalid input.please enter a numerical index.")
```

Result:

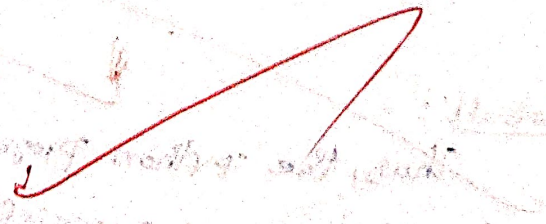
Thus the python program to implement the grades list was sucessfully executed.

output :-

Grades list : [85, 90, 78, 92, 88]

Enter the index of the grade you want to view : 13

"Invalid index. please enter a valid index.



Task No : 9.2

Aim: To write a python program that performs basic arithmetic operations.

Algorithm:-

1. Start the programs
2. Prompts the user to enter two numbers: 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 display 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 numerator
  numerator = float(input("Enter the numerator: "))
  # Prompt the user to enter the denominator
  denominator = float(input("Enter the denominator: "))
# Attempt to perform division
  result = numerator/denominator
except ZeroDivisionError:
  # Handle division by zero error
  print("Error: Division by zero is not allowed.")
except ValueError:
  # Handle invalid input that is not a number
  print("Error: Please enter valid numbers.")
# Call the function to execute the division operation divide_numbers()
```

Result:

Thus the program to calculator program that performs basic arithmetic operations was successfully executed.

Output :-

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

Task 9.3 Determine eligibility to vote

Aim: To write a python application to determine if a pelson is eligible to vote based on their age.

Algorithm:-

1. Define the custom exception
2. prompt the user for input
3. Check if 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-defined exceptions
class Invalid Age 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 Expection
  else:
    Print ("Eligibe to vote")
  except InvalidAge Exception:
    Print ("Exception occured: Invalid Age")
```

V [illegible] TECH - CSE	
EX NO.	9
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	
SIGN WITH DATE	15

Result:

Thus the program application to determine if a person is eligible to voted based on their age

Output :

Enter a number: 117
Exception occurred : Invalid Age