Task 9

Implement Exceptions and Exceptional handling in python

Aim: To implement exceptions and Exceptional handling in python.

Algorithm:

1) Start the program
2) Initializes a list of grades (
3) Prompts the use to enter the index of the grade they with to view
4) Attempts to displays the grade of the specified index.
5) If the index is out of range. catches the Index error and prints an error message "Invalid index. please 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 input is not an integer
print ("Invalid input. Please enter a numerical index")
```

Result: Thus the program for Implement Exception and handling in python.

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
```

Task 9.2 you are developing a python calculate program the perform that basic arithmetic operations. One of the key functionalities is to divide two numbers entered by the user.

Aim: you are developing a python calculate program the perform that 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 numbers: a numerator and a denominator.
3) Attempts to divide the numerator by the denominator.
4) If the denominator is zero, catches the zero DivisionError 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 numerator
 numerator = float (input ("Enter the numerator:"))
#prompt the user to enter the denominator
denominator = float (input ("Enter the denominator:"))
#prompt the Attempt to perform division
        result = numerator / denominator
        print (f" Result : {result}")
  except zero DivisionError:
   # Handle division by zero error
    print("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.
```

Output :

```
Enter a number: 15
exception occurred: InvalidAge
```

Task 9.8

Aim: You are building a python application to determine if a person is eligible to vote base on their age. According to the rules only individuals who are 18 years (or) older are allowed to vote.

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-defined exceptions
 class Invalid Age 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 vote')
except Invalid Age exception:
   print ("Exception occured: Invalid Age")
```

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

Result: Thus the program for Implement Exceptions and exceptional handling is executed and verified successfully.