Task :- 9 Date :- 22/9/25

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 (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 specified 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 grade 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 numeri-
-cal index")
```

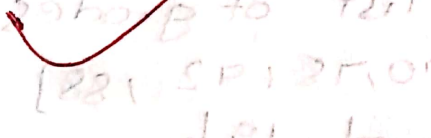
output:-

Enter the numerator:10.

Enter the denominator:0

ERROR!

Error: Division by zero is not allowed.



Problem 9.2 you are developing a python calculator program that performs basic arithmetic operations. one of the key functionality 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 deromirator.
4. if the deromirator 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 numerator.
    numerator = float (input ("enter the numerator:"))
# prompt the user to enter the denaminator:"))
# Attempt to perform division result = numerator/
                                      denominator
  Print (f"Result: {result}")
 except zero division error:
# Handledivision 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_number()
```

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.

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 (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")
```

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