Task 9 :- Implement Exceptions and Exception handling

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

Algorithm:-

1. Start the program.
2. Initializes a list of grades (eg; [85, 90, 78, 92, 88])
3. prompts the users 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 rage, 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 what to view.
try.
  index = int(input ("Enter the index of the grade
  they what 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
  rage print ("Invalid index. please enter a valid
  index.
```

output

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

Enter the index of the grade you want to view: 6

Invalid index. please enter a valid index.

output:

Enter the numerator: 10

Enter the denominator: 0

ERROR!

Groot: Division by zero is not allowed!

except value error:
Handle the case where the input is not an input
integer print ("Invalid input please enter a numeri
cal index:")

problem 9.2:- you are developing a python calculator program that performs basic arithmetic operations. one of key functionalities is to divide two numbers would cause program to crash if not handled properly.

Algorithm:-

1. start the program
2. prompts the user to enter two numbers: a numerator and a denominator
3. Attempts to divide the numeric by the denominator.
4. if the denominator is zero, cathes the zero division error and displays an error messages: "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
print (f "Result: {result}")
except zero division Error:
# 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 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. According to the rules, only individuals who are 18 years (or) older are allowed to vote. To enforce this rule, you decide to create an age below 18 is entered.

Algorithm

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

Program:-

```
# define Python user-defined exceptions
class InvalidAgeException(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 InvalidAgeException
else:
    print("Eligible to vote")
except InvalidAgeException:
print("Exception occured: Invalid Age:")
```

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