Task 9: Implement Exceptional handling in pytho

9.1 You are developing a python program that processes a list of students grades. The program is designed to allow the user to select a grade by specifying an in index number

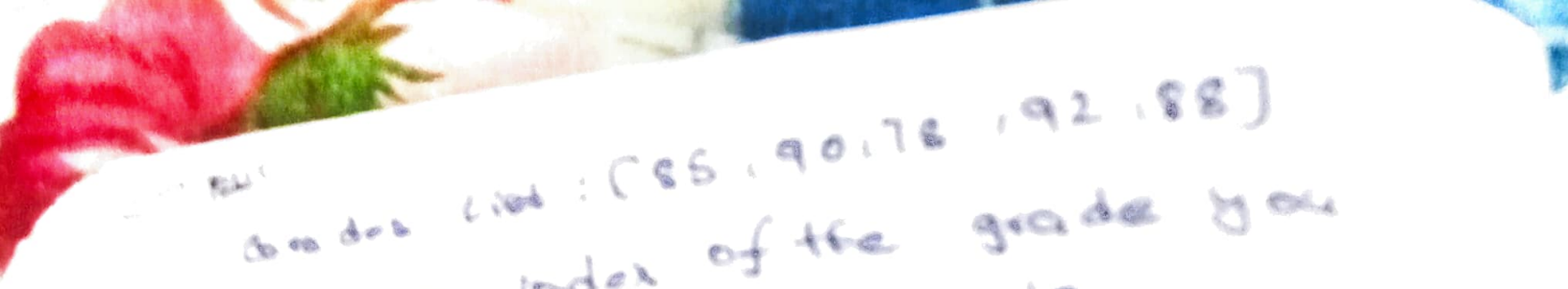
Algorithm

1. start the program
2. Initialize a list of grades
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

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
          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 ValueError:
    # Handle the case where the input
        is not an integer
    print ("Invalid input. Please enter a
        numerical index.")
```



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.

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 numbers: a numerator and a denominator.
3. Attempts to divide the numerator by the denominator.

Program:-

```
# Function to perform division
def devide_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 ZeroDivisionError:
        # Handle division by zero error
        Print("Error: please enter valid numbers.")

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

output:

enter the numerator:10

enter the deromintor:0

ERROR!

Error: Division by zero is not allowed

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

Program:

```
# define python user-defined exceptions
class InvalidAgeException
    "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 InvalideAgeException
    else:
        print("Eligible to vote")
except InvalidAge
    print("Exce
```

Result:- Thus the program for Exception and Exceptional handling is executed and verified Successfully

output:-

```
enter a number:15
Exception occurred : Invalid Age
```