PRACTICAL – EXCEPTION

PROGRAM -1

AIM- Write a python program to create a function to read a file if while opening a file, while reading a file is not found in the system then handle this exception. This function always has to close the file and print the message of closing the file.

CODE-

|  |
| --- |
| Print(“HARSH D ”)  def read\_file(file\_name):      try:          with open(file\_name, 'r') as file:              content = file.read()              print(content)      except FileNotFoundError:          print("File not found.")      finally:          print("Closing the file.")    # Example usage:  file\_name = "harsh.txt"  read\_file(file\_name) |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -2

AIM- Write a python program to divide two numbers and handle if a 0 division error or value error occurs during the division, if there is no exception while dividing two numbers then print the result.

CODE-

|  |
| --- |
| # Solution:  print("HARSH D")  try:  num1 = int(input("Enter the first number: "))  num2 = int(input("Enter the second number: "))  result = num1 / num2  print("Result:", result)  except ZeroDivisionError:  print("Error: Division by zero is not allowed.")  except ValueError:  print("Error: Invalid input. Please enter valid numbers.")  except Exception as e:  print("An error occurred:", str(e))  # Output:  # Enter the first number: 10  # Enter the second number: 0  # Error: Division by zero is not allowed.  # Enter the first number: 10  # Enter the second number: 5  # Result: 2.0 |

OUTPUT:-

|  |
| --- |
|  |

PROGRAM -3

AIM- Write a python program to create custom exceptions to check whether the entered phone number is of 10 digits or not.

CODE-

|  |
| --- |
| # Write a python program to create custom exceptions to check whether the entered phone number is of 10 digits or not.  print("HARSH D")  class PhoneNumberException(Exception):  pass  def check\_phone\_number(phone\_number):  if len(phone\_number) != 10:  raise PhoneNumberException("Phone number must be 10 digits long.")  try:  phone\_number = input("Enter your phone number: ")  check\_phone\_number(phone\_number)  print("Phone number is valid.")  except PhoneNumberException as e:  print(e) |

OUTPUT-

|  |
| --- |
|  |
|  |

PROGRAM -4

AIM- Write a python program to create custom exceptions to check if the employee's salary is above 10,00 or not. If it is above 10,000, then he is eligible for promotion else he isn’t.

CODE-

|  |
| --- |
| # Write a python program to create custom exceptions to check if the employee's salary is above 10,00 or not. If it is above 10,000, then he is eligible for promotion else he isn’t.class SalaryException(Exception):  print("HARSH D")  class SalaryException(Exception):  def \_\_init\_\_(self, message):  self.message = message  # Function to check salary for promotion eligibility  def check\_promotion\_eligibility(salary):  if salary > 10000:  return "You are eligible for promotion"  else:  raise SalaryException("You are not eligible for promotion")  # Example usage  try:  # Assume the salary is provided by the user or retrieved from a database  salary = int(input("Enter your salary: ")) # Prompt the user to enter their salary  message = check\_promotion\_eligibility(salary)  print(message)  except SalaryException as e:  print(e) |

OUTPUT-

|  |
| --- |
|  |
|  |

PROGRAM -5

AIM- Write a python program to create custom exceptions to check the age of citizens. If the age is above 18 years, then he/she is eligible to vote else not.

CODE-

|  |
| --- |
| # Write a python program to create custom exceptions to check the age of citizens. If the age is above 18 years, then he/she is eligible to vote else not.  print("HARSH D")  class AgeError(Exception):  pass  try:  age = int(input("Enter your age: "))  if age < 18:  raise AgeError  else:  print("You are eligible to vote.")  except AgeError:  print("You are not eligible to vote.") |

OUTPUT:-

|  |
| --- |
|  |
|  |