SUMANTH S

AF0363570

PYTHON LAB: 11 EXCEPTION HANDLING

“Exception handling in Python refers to the mechanism used to deal with errors or exceptional situations that may occur during the execution of a program.”

Questions:

1. Write a Python program to handle a ZeroDivisionError exception when dividing a number by zero

Solution:

```
try:
    number1 = int(input("Enter a number: ")) # taking input from the user
    number2 = int(input("Enter a number: ")) # taking input from the user
     division = number1 / number2 # dividing the number1/number2 → (invalid expression)→ number1/0.
except ZeroDivisionError: # if the number is divided by zero catching the ZeroDivisionError exception
    print("Division by zero is not allowed.") # print statement → Division by zero is not allowed
else: # print the division result
    print("The result of the division is", division)
finally: # print the default statement
    print("The program completed successfully.")
```

Output:

```
Enter a number: 2
Enter a number: 0
Division by zero is not allowed.
The program completed successfully.
```

```
Enter a number: 14
Enter a number: 7
The result of the division is 2.0
The program completed successfully.
```

2. Write a Python program that prompts the user to input an integer and raises a ValueError exception if the input is not a valid integer.

Solution:

```python
try:
 number = int(input("Enter an integer:")) # taking input from user
 print("The entered integer is:", number) # print the input if it is number
except ValueError: # catching the exception if the input is other than a number
 print("Enter the input only in numbers") # print the exception statement
finally: # print default statement
 print("The program completed successfully.")
```

Output:

```
Enter an integer:5d
Enter the input only in numbers
The program completed successfully
```

```
Enter an integer:1234445
The entered integer is: 1234445
The program completed successfully.
```

3. Write a Python program that opens a file and handles a FileNotFoundError exception if the file does not exist.

Solution:

```python
try:
  with open('abc.txt', 'r') as file: # Open the file in read mode
    content = file.read() # Read the contents of the file
    print("File contents:") # print File contents
    print(content)
except FileNotFoundError: # if the file is not found using exception FileNotFoundError
  print("File not found. Please make sure the file exists.")
```

Output:

```
File not found. Please make sure the file exists.
```

4. Write a Python program that prompts the user to input two numbers and raises a TypeError exception if the inputs are not numeric

Solution:

```
try:
    number1 = input("Enter the first number: ") # user to input the first number
    number2 = input("Enter the second number: ")  # user to input the second number
    multiplication = number1*number2 # performing multiplication
    print(multiplication)
except TypeError: # print TypeError exception statement
    print("Error: Multiplication cannot perform for one number and string.")
finally: # print default statement
  print("The program executed successfully")
```

Output:

```
Enter the first number: 10
Enter the second number: d
Error: Multiplication cannot perform for one number and string.
The program executed successfully
```