UNIVERSITY OF
BIRMINGHAM
DUBAI

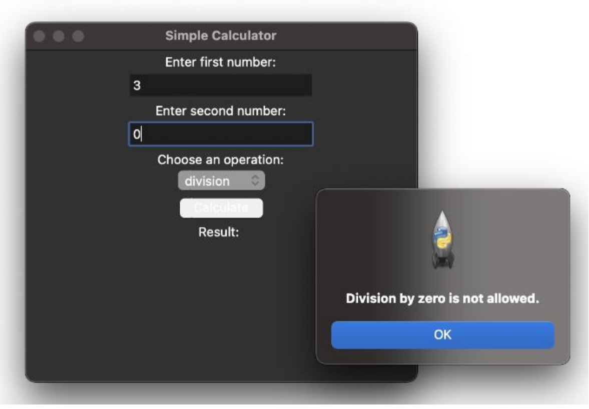
LM Software Workshop 1 (34153, 34182, 34168, 36990)
Lab Exercise Sheet
Week 7 Exception Handling and Tkinter

1. **Write a Python program that uses a try-except-finally block to open and read a file**. The program should handle any `FileNotFoundError` exceptions that occur if the file does not exist, and use the finally block to confirm that the file has been closed.

2. **Create a Python program that handles multiple exceptions**. The program should try to open a non-existent file and read an integer from it, thus potentially causing both `FileNotFoundError` and `ValueError` exceptions.

3. Create a function that asks the user to input their birth year and calculates their age. If the user enters an invalid value that cannot be converted to an integer, handle the exception by asking the user to input a valid year.

4. **Write a Python program that acts as a simple calculator**. The program should ask the user for two numbers and an operation (addition, subtraction, multiplication, or division). Use try-except blocks to handle any exceptions that might occur, such as `ZeroDivisionError` (when the user tries to divide by zero) or `ValueError` (when the user inputs something that's not a number). *- The user is expected to use the command line to handle the program however, as an optional task, convert your program to a GUI to be able to present a graphical user interface using Tkinter where the user can input two numbers and select an operation (addition, subtraction, multiplication, or division)—something like the example below.*



Simple Calculator

Enter first number: 3

Enter second number: 0

Choose an operation: division

Result:

Division by zero is not allowed.

5. You have a dictionary of student names and their corresponding grades. Write a function that takes a student's name as input and returns their grade. If the student's name is not found in the dictionary, handle the exception and inform the user.