National University of Computer and Emerging Sciences

Lab Exercise 03
DL2001-Introduction to Data Science Lab

Course Instructor	Ms. Mariam Nasim
Lab Instructor(s)	Ms. Rida Amir
Section	BDS-3A
Semester	Fall 2025

Department of Data Science

FAST-NU, Lahore, Pakistan

Exercise

1. Develop a Python program that performs basic mathematical operations (addition, subtraction, multiplication, division) and handles errors like **division by zero** and **invalid inputs**.
    - Define four functions for addition subtraction, multiplication, and division, each taking two numbers and returning the result.
    - Request two numbers from the user, using try-except to validate they are numeric and re-prompting if not.
    - Ask the user for the desired operation, using their input to call the corresponding math function
    - Specifically for division, include a try-except block to handle division by zero, displaying an error message without crashing
    - Display the operation's outcome or an error message, then ask if they'd like another calculation, repeating the process or ending based on their response

    Test your program for each operation and with invalid inputs to ensure Exception handling works as expected. Create one code block for each operation

2. Design a code which reads text from the file **"Alphabets.txt"** and stores its data in reverse order in another file. For this you may upload the given text file on Google Colab's session and define the path as:

```
source_file_path= '/content/Alphabets.txt'
```

    The same convention can be followed for defining path of the resultant file (reversed text file).

3. Write a Python function that reads numbers from a file, calculates their sum, and saves the result to another file. Use exception handling to deal with potential errors, such as missing files (`FileNotFoundError`), non-numeric values in the file (`ValueError`), and permission issues (`PermissionError`) when writing the result file.
    - Define a function named sum_file that accepts two parameters: the path of the input file and the path of the output file.
    - Inside the function, attempt to open the input file in read mode. Use a try-except block to catch and handle a missing file error **(FileNotFoundError).**

- Read each line from the file, convert it to a number, and add it to a running total. Use a try-except block to catch and handle non-numeric values **(ValueError)**, skipping them or displaying a warning.
- Open the output file in write mode and save the sum there. Use try-except to handle potential issues when opening or writing to the file, such as permission errors **(PermissionError)**.

Test your function with various scenarios, including a correct input file, a non-existent file and a file with some non-numeric values

4. You are developing a simple system to manage books in a library. You need to create a class that represents a Book and allows users to check out or return books.
    a. Create a class called Book with the following attributes:
        i. title: The title of the book (string)
        ii. author: The author of the book (string)
        iii. year: The year the book was published (integer)
        iv. available: A boolean indicating whether the book is available for borrowing (default should be True)
    b. Implement the following functions:
        i. `get_info(self)`: Returns a string with the book details in the format: "Title (Year) by Author - Available" or "Title (Year) by Author - Checked Out"
        ii. `checkout(self)`: If the book is available, set available to False and print "You have checked out 'Title'." If not available, print "Sorry, 'Title' is currently not available."
        iii. `return_book(self)`: If the book is not available, set available to True and print "You have returned 'Title'." If already available, print "This book was not checked out."
    c. Test your code using the following:

```
book1 = Book("The Hobbit", "J.R.R. Tolkien", 1937)
print(book1.get_info())           # "The Hobbit (1937) by J.R.R. Tolkien -
Available"
book1.checkout()                  # "You have checked out 'The Hobbit'."
print(book1.get_info())           # "The Hobbit (1937) by J.R.R. Tolkien -
Checked Out"
book1.checkout()                  # "Sorry, 'The Hobbit' is currently not
available."
book1.return_book()               # "You have returned 'The Hobbit'."
book1.return_book()               # "This book was not checked out."
```