**Exercise 1: Handling Files with Python**

**Objectives:**

* Read from and write to files in Python.
* Differentiate between binary and text files.

**Business Scenario:**

You are creating a file-based contact management system. The system should store contacts in a file and allow users to read, add, and remove contacts.

**Tasks:**

1. **Text Files:**
   * Create a new Python file named **contact\_management.py**.
   * Write functions to read from and write to a text file.
   * Add, remove, and display contacts stored in the text file.
2. **Binary Files:**
   * Write functions to handle binary files.
   * Save and load contact information in a binary format.
3. **User Interaction:**
   * Capture user input to add and remove contacts.
   * Display the list of contacts stored in the file.
4. **Error Handling:**
   * Implement error handling for file operations.
   * Ensure the program gracefully handles missing files or corrupted data.

**Exercise 12: Error Handling**

**Objectives:**

* Handle errors and exceptions in Python.

**Business Scenario:**

You are developing a financial application that processes transactions. The application should handle various errors that may occur during processing, such as invalid input or file not found.

**Tasks:**

1. **Error Handling:**
   * Create a new Python file named **transaction\_processing.py**.
   * Use try-except blocks to handle potential errors during transaction processing.
   * Implement custom error messages for different types of errors.
2. **Validation:**
   * Validate transaction data before processing.
   * Prompt the user to correct any invalid data.
3. **Logging Errors:**
   * Write a function to log errors to a file.
   * Ensure all exceptions are logged with a timestamp.
4. **User Feedback:**
   * Provide meaningful feedback to the user when errors occur.
   * Ensure the application continues running after handling an error.