**Lab 1: Opening and Closing Files in Python**

1. Write a Python program to:
   * Open a text file named sample.txt
   * Read and print its contents
   * Close the file properly using close()
2. Modify the above program to use the with statement.

**Lab 2: Reading a File - read(), readline(), readlines()**

1. Create a text file with multiple lines.
2. Write a Python program to:
   * Read the entire file using read()
   * Read line by line using readline()
   * Read all lines into a list using readlines()
3. Compare the outputs of these methods.

**Lab 3: Writing to a File (w mode)**

1. Write a program to:
   * Open a file in w mode
   * Write multiple lines of text into it
   * Close and reopen the file to verify its contents

**Lab 4: Appending to a File (a mode)**

1. Open the previous file in a mode.
2. Append additional lines to it.
3. Verify that the old content is preserved.

**Lab 5: Read & Write Using r+ Mode**

1. Open the file in r+ mode.
2. Read its contents and append a new line.
3. Observe the file pointer position before and after writing.

**Lab 6: Read & Write Using a+ Mode**

1. Open a file in a+ mode.
2. Append new text and read the entire content.
3. Explain the difference between r+ and a+.

**Lab 7: Writing a List to a File**

1. Create a list of strings.
2. Write the list to a file using writelines().
3. Read the file and print its contents.

**Lab 8: Reading and Writing JSON Files**

1. Create a Python dictionary with sample data.
2. Convert it into JSON format and write it to a file.
3. Read the JSON file and convert it back into a dictionary.

**Lab 9: Transforming Python Objects into JSON**

1. Create a Python class Person with attributes name and age.
2. Create an instance and serialize it to a JSON file.
3. Deserialize it back into a Python object.

**Lab 10: Transforming Python Objects into CSV**

1. Create a list of dictionaries representing student data.
2. Write it into a CSV file using csv.DictWriter().
3. Read the CSV file and display its contents.

**Lab 11: Reading CSV Data**

1. Create a CSV file with some sample rows.
2. Read it using:
   * csv.reader()
   * csv.DictReader()
3. Compare the outputs of both methods.

**Lab 12: Writing CSV Data**

1. Write a program to:
   * Open a CSV file in write mode
   * Write multiple rows of data
   * Close and read the file to verify

**Lab 13: Handling Different CSV Dialects**

1. Create CSV files using different delimiters (comma, tab, semicolon).
2. Read them by specifying the correct csv.Dialect.

**Lab 14: Appending Data to CSV Files**

1. Open an existing CSV file.
2. Append new rows without overwriting existing data.
3. Read and print the updated file.

**Lab 15: Handling JSON Nested Data**

1. Create a JSON file containing nested data (list of dictionaries).
2. Read and parse the nested data in Python.
3. Extract and print specific values.

**Lab 16: Reading and Writing Excel Files (pandas)**

1. Create a Pandas DataFrame.
2. Write it to an Excel file.
3. Read it back into a DataFrame.

**Lab 17: Converting JSON Data to CSV**

1. Read a JSON file containing a list of dictionaries.
2. Convert it into a CSV file.

**Lab 18: Converting CSV Data to JSON**

1. Read a CSV file.
2. Convert it into JSON format and write to a file.

**Lab 19: Object Serialization - YAML**

1. Create a Python dictionary.
2. Serialize it into YAML format and write it to a file.
3. Read and deserialize the YAML file.

**Lab 20: Comparing JSON, CSV, and YAML Serialization**

1. Serialize the same Python dictionary into JSON, CSV, and YAML formats.
2. Compare their readability and storage efficiency.