

Advanced Programming Practice Questions (Week-2)

1. Create a TXT file and write the numbers from 1 to 100 each in a new line.
2. Write a function that reads the file you have created in Q1 and find the sum of the numbers write the result in the last (new) line in the file.
3. Create a file that you can both read and write from simultaneously. Write the following sentences in it (I LOVE PROGRAMMING!) then read it out of the file. Make sure to print out each word on a separate line.
4. Read the file called “try.txt” as file and find the output of the following;

```
print(file.read(4))  
print(file.read(5))  
print(file.read())
```
5. Write a Python program that counts the number of “me” and “my” (case-insensitive) words present in the given file “story.txt”.
6. Write a Python program that copies one text file to another text file, omitting any lines that begin with # (Use the given file called "mixed_lines.txt")
7. Write a function which;
 - a. Opens the file "words.txt"
 - b. Finds frequency of each letter
 - c. Writes them into a new file called "freq_words.txt" as {frequency: [letter1, letter2,...]} pairs
8. Write a function called "sumAll" that reads the given file "interger.txt" containing only integers and return the sum, min, and max of all the integers.
9. Write a program that reads the given file “abc.txt” and creates a new file with the lines in reversed order (i.e., the first line in the old file becomes the last one in the new file.)
10. Write a function that uses the file given in the previous question "abc.txt" to have the file "abc.numbered.rev". (You have to make it reversed and line numbered)

11. Using the given “university_data.json” file, implement a code for the following tasks.

- a. Find the total number of faculties in the university?
- b. List all the course codes and their corresponding course names.
- c. Find the faculty with the most courses offered.
- d. Determine the total number of credits for all courses in the Faculty of Engineering
- e. Find the course with the highest number of registered students.
- f. List all courses offered on Mondays.
- g. Find the lecturer for a specific course (e.g., EE301).
- h. Determine if a course with a specific course code (e.g., CS101) exists.
- i. List all courses offered by a specific lecturer (e.g., Dr. John Smith).
- j. Find the number of courses offered on Wednesdays.

12. Write a Python script that creates a JSON file named "students.json" and adds sample data representing information about three students. Each student should have attributes for their name, age, and grade. Then by using Error Handling Methodology, the script should print a message confirming the successful creation of the JSON file with the sample data, if there is an error, the program shouldn't crash.

Sample Data:

Student 1: Name - "John Smith", Age - 18, Grade – 12

Student 2: Name - "Emily Johnson", Age - 17, Grade – 11

Student 3: Name - "Michael Brown", Age - 16, Grade – 10

13. Write a Python script that creates an Excel file named "students.xlsx" and adds sample student data to it. The sample data includes the names, ages, and grades of three students. Your script should handle any errors that may occur during the file creation process and print an appropriate error message.

Sample Data:

Student 1: Name - "John Smith", Age - 18, Grade – 12

Student 2: Name - "Emily Johnson", Age - 17, Grade – 11

Student 3: Name - "Michael Brown", Age - 16, Grade – 10

14. Using the given “school_data.xlsx” file, implement a code for the following tasks.

- a. How many students are there in the dataset?
- b. What is the average age of the students?
- c. How many male and female students are there?
- d. Who is the oldest student in the dataset? What is their age?
- e. What is the highest GPA among the students?
- f. List the names of students with a GPA above 3.5.
- g. Calculate the total GPA of all female students.
- h. How many students have a GPA below 2.5?
- i. Calculate the median age of the students.
- j. What is the most common age among the students?