“Important Questions and Lessons on Python – Mod 05” _Mohiuddin Khan

Below are the answers to each question with definitions and short code examples. I cited the following class notes, reference textbook, and authoritative online references for completeness:

(a) **Course Module Notes** (Mod05_Notes)

(b) **Textbook** (*Python Programming for the Absolute Beginner, 3rd ed.*)

(c) **Official Docs / Reputable Guides:** Built-in types, expressions/operators, exceptions, built-in functions (incl. type()), PEP 8 style

Questions and Answers:

1) What is the difference between a List and a Dictionary?

A List is an ordered collection of items where each item is accessed using a number index (0, 1, 2…). A Dictionary is an unordered collection of key–value pairs, where each value is accessed using a key name (like "Name" or "Email").

Example:

- List → ["Bob", "Sue", "Tom"]
- Dictionary → {"Name": "Bob", "Email": "bob@gmail.com"}

Module 05 Notes show: A list stores whole dictionaries like rows in a table, while a dictionary stores labeled data such as "ID", "Name", and "Email".

2) What is the difference between an Index and a Key?

- An Index is a number used to access a position in a list.
  Example: students[0]
- A Key is a string used to access a value in a dictionary.
  Example: student["Name"]

Module Notes explain that keys are case-sensitive and act like column names in a spreadsheet.

3) How do you write data to a file from a Dictionary?

We loop through each dictionary in a list and write the values to a file using the keys.

Example from Module 05:

```
file = open(FILE_NAME, 'w')
```

```
for each_row in table:
    file.write(f'{each_row["ID"]}, {each_row["Name"]}, {each_row["Email"]}\n')
file.close()
```

This writes each row as a comma-separated line.

4) How do you read data from a file into a Dictionary?

We read the file line-by-line, split each line, then assign fields into a dictionary using keys.

Example from Module:

```
file = open(FILE_NAME, "r")
for each_row in file:
    data = each_row.split(",")
    row = {"ID": data[0], "Name": data[1], "Email": data[2].strip()}
    table.append(row)
file.close()
```

This converts each CSV line into a dictionary and appends it to a list.

5) What is a JavaScript Object Notation (JSON) file?

A JSON file is a text file format that stores data using key–value pairs similar to Python dictionaries. JSON is used for data exchange because it's easy for humans and machines to read. Module Notes explain that JSON is related to dictionaries and requires double-quotes around keys.

6) What does Python's json module do?

Python’s json module allows us to:

- Convert a dictionary → JSON string (json.dumps)
- Save dictionaries to a JSON file (json.dump)
- Load JSON files into Python dictionaries (json.load)

This module makes file handling much easier because JSON format matches Python dictionaries.

7) What is Structured Error Handling?

Structured error handling is a programming method used to **detect and handle errors** without crashing the program.
It uses:

- Try
- Except
- Else
- Finally (optional)

This helps our program run safely even when something unexpected happens (like dividing by zero or entering a wrong file name). Both the Module Notes and the textbook describe using try–except to prevent crashes.

8) Why is error handling using Try-Except recommend?

Because it:

- Prevents our program from stopping unexpectedly
- Allows us to give user-friendly error messages
- Helps us handle mistakes safely (wrong input, missing files, bad conversions)

Module Notes emphasize that Try-Except helps avoid errors that would normally crash our program.

9) What are two common locations for storing and sharing code files?

According to Module 05 Notes, two common storage options are:

1. Network file sharing (shared drives at school/work)
2. Cloud storage (Google Drive, OneDrive, iCloud)

Module Notes list both under “Managing Code Files.”

10) What is GitHub, and why is it used?

GitHub is an online platform used to store, share, and collaborate on code projects.
It allows us to:

- Save code online (like cloud storage for programmers)
- Track changes using Git version control
- Work with teams on the same project
- Publish and share our work professionally