Module 15) Advance Python Programming

Printing on Screen

**Theory:**

Introduction to the print() function in Python.

* Ans- **print() Function:** Displays output on the screen.

Formatting outputs using f-strings and format().

* Ans -
  + **f-strings** → Embed expressions directly within string literals.
  + **format()** → Allows formatted output using placeholders.

Reading Data from Keyboard

**Theory:**

Using the input() function to read user input from the keyboard.

Ans- input() reads user input as a string.

* Example: name = input("Enter your name: ")

Converting user input into different data types (e.g., int, float, etc.).

* Ans - Use type conversion functions to convert input:
  + int(input()) → Converts input into an integer.
  + float(input()) → Converts input into a float.

Opening and Closing Files

**Theory:**

Opening files in different modes ('r', 'w', 'a', 'r+', 'w+').

### Ans -Opening files in different modes

* **'r' (Read mode)** → Opens an existing file for reading.
* **'w' (Write mode)** → Creates a new file or overwrites an existing file.
* **'a' (Append mode)** → Adds content to an existing file.
* **'r+', 'w+'** → Read and write modes.

Using the open() function to create and access files.

* Ans - Syntax: file = open("filename.txt", "mode")

Closing files using close().

* Ans - Always close files using file.close() to free system resources.

4. Reading and Writing Files

**Theory:**

Reading from a file using read(), readline(), readlines().

### Ans -Reading from a file

* **read()** → Reads the entire content of a file.
* **readline()** → Reads one line at a time.
* **readlines()** → Reads all lines and returns a list.

Writing to a file using write() and writelines().

* Ans-  **write("text")** → Writes a string to the file.

 **writelines(["line1", "line2"])** → Writes multiple lines at once.

Exception Handling

**Theory:**

Introduction to exceptions and how to handle them using try, except, and finally.

Ans -  An exception is an error that occurs during program execution.

 Example: Division by zero (ZeroDivisionError), invalid input (ValueError)

**Handling exceptions using try, except, and finally**

* **try block:** Contains code that may cause an error.
* **except block:** Handles the error.
* **finally block:** Executes code regardless of errors.
* Understanding multiple exceptions and custom exceptions.

Ans –  Handle multiple exceptions using multiple except blocks.

*  Create custom exceptions using raise and define exception classes.

Class and Object (OOP Concepts)

**Theory:**

Understanding the concepts of classes, objects, attributes, and methods in Python.

Ans -Understanding Classes and ObjectsAttributes and Methods

* **Class:** A blueprint for creating objects.
* **Object:** An instance of a class.
* **Attributes:** Variables that belong to a class.
* **Methods:** Functions inside a class that operate on objects.

Difference between local and global variables

* Ans - **Local Variable:** Defined inside a function and accessible only within that function.
* **Global Variable:** Defined outside all functions and accessible throughout the program.

Inheritance

**Theory:**

Single, Multilevel, Multiple, Hierarchical, and Hybrid inheritance in Python.

### Ans -Types of Inheritance

* **Single Inheritance:** One parent class and one child class.
* **Multilevel Inheritance:** A class inherits from another derived class.
* **Multiple Inheritance:** A child class inherits from multiple parent classes.
* **Hierarchical Inheritance:** Multiple child classes inherit from one parent class.
* **Hybrid Inheritance:** A combination of multiple inheritance types.

Using the super() function to access properties of the parent class.

* Ans - The super() function allows calling methods of the parent class in a child class.

Method Overloading and Overriding

**Theory:**

Method overloading: defining multiple methods with the same name but different parameters.

### Ans -Method Overloading

* Defining multiple methods with the same name but different parameters.
* Python does not support it directly but can be achieved using default arguments.

Method overriding: redefining a parent class method in the child class.

### Ans -Method Overriding

* Redefining a method from the parent class in a child class.
* The child class method takes precedence over the parent class method.

.9. SQLite3 and PyMySQL (Database Connectors)

**Theory:**

Introduction to SQLite3 and PyMySQL for database connectivity.

Ans -  **SQLite3** → A lightweight database for local storage.

 **PyMySQL** → A MySQL database connector for Python.

Creating and executing SQL queries from Python using these connectors.

* Ans - Connect to the database, create tables, insert data, retrieve data using SQL queries.

10. Search and Match Functions

**Theory:**

Using re.search() and re.match() functions in Python’s re module for pattern matching.

**Ans -**  **re.search()** → Searches for a pattern anywhere in a string.

 **re.match()** → Checks if the pattern appears at the beginning of the string.

Difference between search and match.

Ans -  **search()** scans the entire string.

 **match()** only checks the beginning of the string.