

**Name:** Paramprakash Makwana

## **Module 15) Advance Python Programming**

### 1.1) Introduction to the print() function in Python.

Ans:- print() is used to display output on the screen. It can print text, numbers, variables, and expressions. Multiple values can be printed by separating them with commas. It is one of the most commonly used functions in Python.

### 1.2) Formatting outputs using f-strings and format().

Ans:- f-strings allow inserting variables directly inside strings using {}. The format() method also formats strings by replacing placeholders with values. f-strings are faster and easier to read. Both help in creating dynamic output.

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

Ans:- input() is used to take user input from the keyboard. It always returns data as a string. The entered value can be stored in a variable. It helps make programs interactive.

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

Ans:- Since input() returns string data, conversion is needed for calculations. Functions like int(), float(), and str() are used for type conversion. This process is called type casting. It ensures correct data type usage.

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

Ans:- 'r' is used to read files, 'w' writes and overwrites files, and 'a' adds data without deleting existing content. 'r+' allows reading and writing without deleting data. 'w+' allows reading and writing but removes old content. Modes define how a file is accessed.

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

Ans:- open() is used to create or open a file in Python. It requires the file name and mode as parameters. It returns a file object used for file operations. It is the first step in file handling.

1.7) Closing files using close().

Ans:- close() is used to close an opened file. It releases system resources and saves changes properly. Not closing files may cause data loss or memory issues. It is good practice to close files after use.

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

Ans:- read() reads the entire file content. readline() reads one line at a time. readlines() reads all lines and stores them in a list. These methods help access file data in different ways.

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

Ans:- write() writes a single string to a file. writelines() writes multiple strings from a list. These methods add or replace file content based on the mode. They are used to store data in files.

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

Ans:- Exceptions are errors that occur during program execution. try is used to test code, except handles errors, and finally runs code regardless of errors. It prevents program crashes. It improves program reliability.

1.11) Understanding multiple exceptions and custom exceptions.

Ans:- Multiple exceptions allow handling different errors separately using multiple except blocks. Custom exceptions are user-defined errors created using classes. They help control program behavior. They make error handling more specific.

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

Ans:- A class is a blueprint for creating objects. Objects are instances of a class containing data and functions. Attributes store data, and methods define behavior. This supports object-oriented programming.

1.13) Difference between local and global variables.

Ans:- Local variables are declared inside functions and accessible only within them. Global variables are declared outside functions and accessible throughout the program. Local variables have limited scope. Global variables have wider accessibility.

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

Ans:- Single inheritance uses one parent class. Multilevel inheritance involves inheritance across multiple levels. Multiple inheritance allows a class to inherit from multiple parents. Hierarchical inheritance has multiple child classes from one parent, and hybrid is a combination of these types.

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

Ans:- super() is used to call parent class methods and constructors. It helps avoid rewriting code. It supports inheritance and improves code reuse. It is commonly used in method overriding.

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

Ans:- Method overloading means using the same method name with different parameters. Python supports it using default arguments or variable-length arguments. It improves code flexibility. It allows methods to perform similar tasks differently.

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

Ans:- Method overriding occurs when a child class provides its own version of a parent class method. It is used to change or extend parent class behavior. The method name remains the same. It supports runtime polymorphism.

1.18) Introduction to SQLite3 and PyMySQL for database connectivity.

Ans:- SQLite3 is a lightweight, file-based database built into Python. PyMySQL connects Python with MySQL database servers. Both allow storing and managing structured data. They help perform database operations from Python.

1.19) Creating and executing SQL queries from Python using these connectors.

Ans:- SQL queries are executed using database connectors like SQLite3 or PyMySQL. Commands such as `execute()` and `commit()` are used. They allow inserting, updating, deleting, and retrieving data. This connects Python programs with databases.

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

Ans:- `re.search()` finds a pattern anywhere in a string. `re.match()` checks only at the beginning of the string. Both are part of the `re` module. They are used for pattern matching and validation.

1.21) Difference between search and match.

Ans:- `re.search()` scans the entire string to find a match. `re.match()` checks only the start of the string. `search()` is more flexible. `match()` is stricter and used when pattern must start from the beginning.