

Que. Introduction to the print() function in Python.

Ans.

Print() function is built-in function python.

Using print function, we can print statement and input from a user on console screen.

Que. Formatting outputs using f-strings and format().

Ans.

Both are python built-in function.

We can format our output using these functions.

We can take input from user and place it in print function using these functions.

Using {} we can define place. {} is placeholder.

```
name = "Vishwajeetsinh"
```

```
age = 24
```

```
print("My name is {} and my age is {}".format(name,age))
```

```
print(f"My name is {name} and my age is {age}")
```

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

Ans.

Input() is built-in function in python.

Using this function we can take input from user and print it on console screen.

By default it return string .

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

Ans.

For convert any data into another data type we use type casting.

Ex.

```
A = 10
```

```
Print(int(a))
```

```
Print(float(a))
```

```
Print(str(a))
```

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

Ans.

Using file handling in python we can read and write in file.

“r” : read mode : it will read the file, file must exist.

“w” : write mode : using write mode we can write in file, if already exist it will overwrite, otherwise it will create new one.

“a” : append mode : it not overwrite in file, it will add data to file.

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

Ans.

Using open() function, it will open a file and the apply an operation.

Ex.

```
f = open("filename","r")
```

```
Print(f.read())
```

```
f.close()
```

Que. Closing files using close().

Ans:

When we open a file for any operation after completing operation we have to close file using close() function.

Ex.

```
f = open("filename","r")
```

```
Print(f.read())
```

```
f.close()
```

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

Ans.

read(): Reads the entire content of the file.

Readline(): read the next line of file, if apply multiple times it will return next line each time.

Readlines():read all lines from file and return it in list of string.

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

Ans.

Write(): write a string into a file.

Writelines(): it is used when you have to add multiple lines at once. It takes list of string and add it without line break. If you want new line use “\n” at end of string.

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

Ans.

Using exception handling we can manage exception in program so that it does not crash.

Instead of letting program stopped when error occurs , we can handle the error gracefully.

Try:

You write the code that might cause an exception inside a try block.

Except:

If an exception occurs within the try block, the code in the except block is executed. You specify the type of exception you want to catch.

Finally:

The finally block is executed no matter what, whether an exception occurs or not.

Que. Understanding multiple exceptions and custom exceptions.

Ans.

Multiple exception:

When we write a block of code, it might raise different types of exceptions, you can handle them using multiple except clauses. This allows you to respond differently depending on the type of exception that was raised.

Custom exception:

Sometimes, the built-in exceptions are not sufficient for your needs, and you may want to define your own exception types. This is done by creating a new class that inherits from the built-in exception class.

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

Ans.

Class: class is a blueprint for creating object.

Object : object is an instance of class.

Attributes: attributes are variable that hold data for the object created from the class.

Methods: methods are functions that define the behavior of the class.

Que. Difference between local and global variables.

Ans.

Local variable	Global variable
Define inside a function	Define outside a function or top of code.
Access within the function	Access anywhere in program
Exist during function execution	Exist for the during program execution
Do not offer data sharing	Offers data sharing

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

Ans.

One class derived properties of another call its called inheritance.

Inheritance provide reusability and reduce our code.

Single inheritance:

One child class inherits from one parent class.

Multi-level inheritance:

A class inherits from another class, which itself inherits from a third class.

Multiple inheritance:

One child class inherits from more than one parent class.

Hierarchical inheritance:

Multiple child classes inherit from a single parent class.

Hybrid inheritance:

A combination of multiple types of inheritance.

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

Ans.

This function is used to give access to properties of parent class. It returns an object that represent the parent class.

Que. Method overloading.

Ans.

Two or more method have same name but different parameters it called method overloading.

By default, python does not support method overloading.

Que. Method overriding.

Ans.

Method overriding means child class has a method with the same name as a method in parent class.

When we call method of child class the child class's method runs not parent class's.

Que. Introduction to SQLite3 and PyMySQL for database connectivity.

Ans.

SQLite3:

SQLite3 is a lightweight, serverless, self-contained relational database.

It stores data in a single file on your system.

Ideal for small to medium projects, desktop apps, or testing.

No need to install a separate database server.

Supports SQL queries (CREATE, INSERT, SELECT, UPDATE, DELETE).

Lightweight and fast.

Python has a built-in module called sqlite3.

PyMySQL:

PyMySQL is a Python library for connecting to MySQL/MariaDB databases.

Works with MySQL server.

Supports transactions, multiple users, and remote access.

Ideal for web applications and large projects.

Install PyMySQL : `pip install pymysql`

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

Ans:

Re.search(): search for pattern anywhere in string.

Re.match(): check if pattern match at beginning of the string.

Que. Difference between search and match.

Ans.

Both are used for searching for a string but re.match() searches only at the beginning of a string. If a match is found, it returns the match object; otherwise, it returns None. re.search() searches the entire string for a match. If a match is found, it returns the match object; otherwise, it returns None.

