**Advanced Python Programming**

(Module-15)

**Assignment**



## SUBMITTED BY:

PRIYA PAREKH

## SUBMITTED TO:

CHINMAYEE MAM

BACKEND DEVELOPMENT TOPS TECHNOLOGIES

1. Printing on Screen

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

Answer:

The print() function in Python is a built-in function used to display output to the console, and it's a fundamental tool for debugging and presenting information to the user. It takes any number of arguments, which are converted to strings and separated by spaces before being printed to the screen, with a newline character added at the end by default.

Syntax:

print(“I love Python”)

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

Answer:

F-strings are a way to embed expressions inside string literals, using curly braces {}. They are prefixed with the letter f or F.

F-strings provide a concise and readable way to format strings.

Syntax: print(f”My Name is {name}.”}

The format() method is a string method that allows you to format strings by placing curly braces {} as placeholders in the string and calling the format() method on the string.

The format() method allows for more complex formatting scenarios, such as specifying the order of arguments and using named placeholders.

Syntax: print(”My Name is {}.”.format(name)}

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

Answer:

The input() function is a built-in function in Python that allows you to take input from the user via the keyboard. It reads a line of text the user enters and returns it as a string.

Syntax: user\_input = input(prompt)

Example:

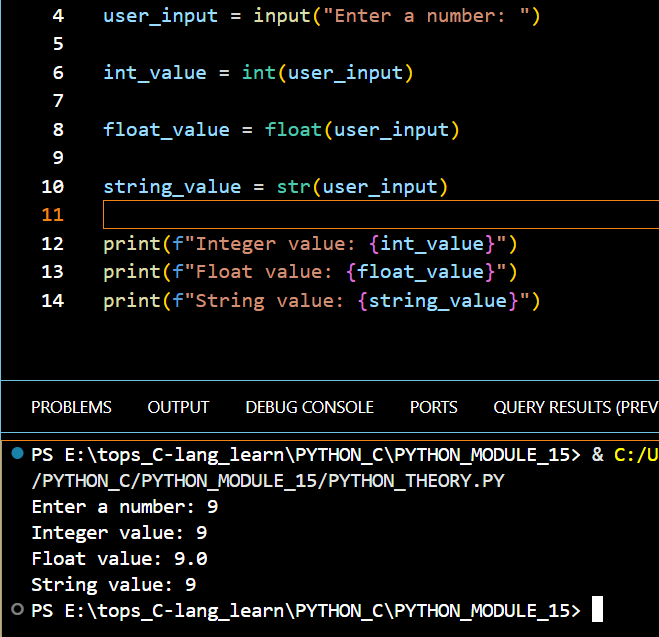
name = input("Please enter your name: ")

print(f"Hello, {name}!")

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

Answer:

Getting user input and converting into str, int, and float.



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

Answer:

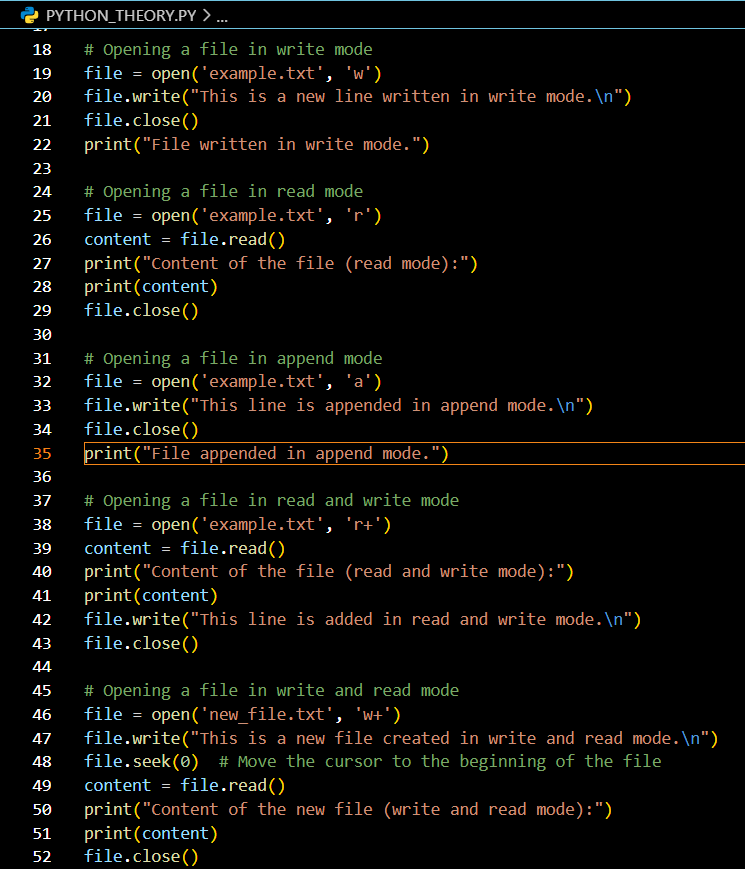
‘r’ : This mode opens file in Read Mode.

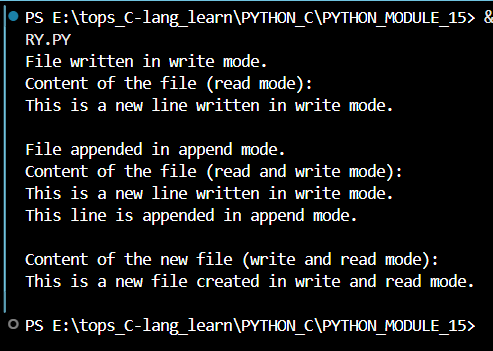
‘w’ : This mode opens file in Write Mode.

‘a’ : This mode opens file in Append Mode.

‘r+’ : This mode opens file in Read and Write Mode.

‘w+’ : This mode opens file in Write and Read Mode.

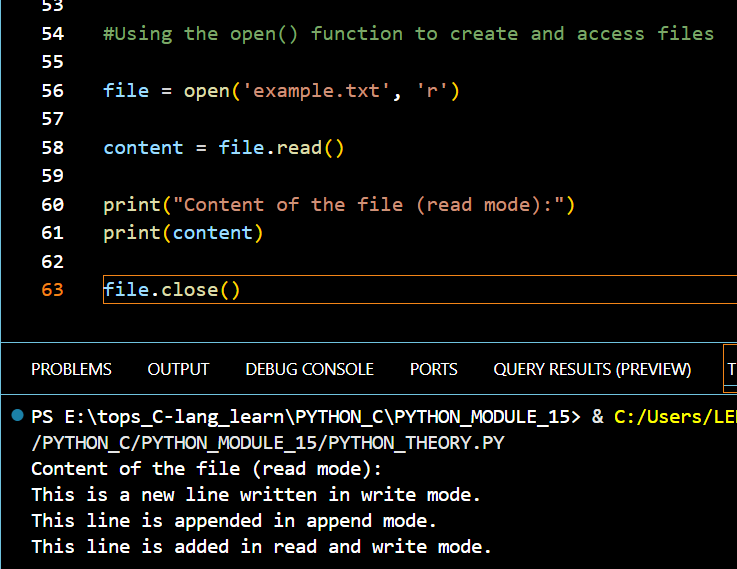




Que. The open() function in Python is used to create and access files.

Answer:

The open() function in Python is used to create and access files.



Que. Closing files using close().

Answer:

call f.close() to close it and free up any system resources taken up by the open file.

Syntax: file\_name.close()

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

Answer:

Methods:

* **read()**: This method reads the entire content of a file and returns it as a single string. It is useful when you want to load the whole file at once, but it may not be efficient for very large files.
* **readline()**: This method reads one line from the file each time it is called, returning it as a string. It is useful for processing files line by line, allowing you to handle large files without loading the entire content into memory.
* **readlines()**: This method reads all lines from the file and returns them as a list of strings, each representing a line. It is convenient for iterating over multiple lines at once but can consume more memory for large files.

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

Answer:

* Use write() for writing single strings or lines to a file.
* Use writelines() for writing multiple strings from an iterable to a file, ensuring to manage line breaks as needed.

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

Answer:

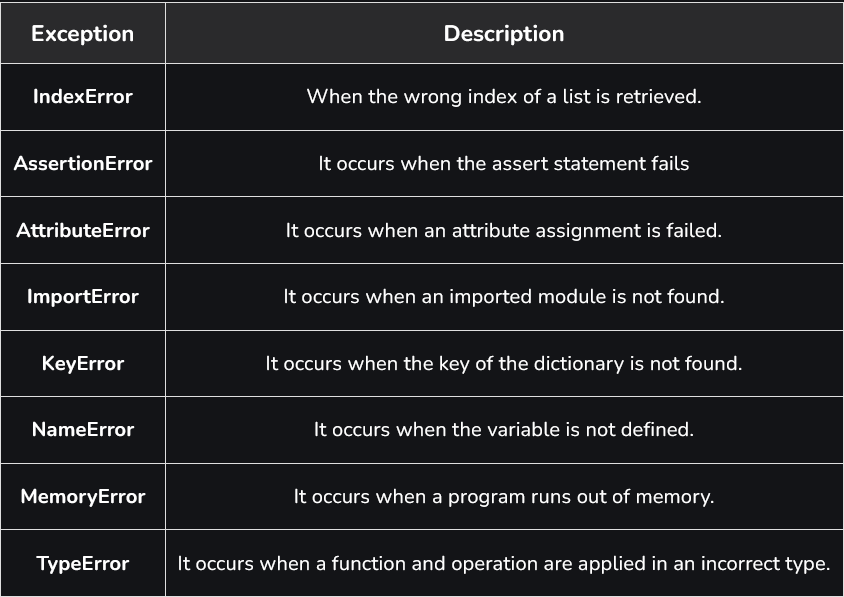
Python Exception Handling handles errors that occur during the execution of a program. Exception handling allows to respond to the error, instead of crashing the running program. It enables you to catch and manage errors, making your code more robust and user-friendly.

Handling Exception:

* **Try Block**: lets us test a block of code for errors. Python will “try” to execute the code in this block. If an exception occurs, execution will immediately jump to the except block.
* **Except Block:** enables us to handle the error or exception. If the code inside the try block throws an error, Python jumps to the except block and executes it. We can handle specific exceptions or use a general except to catch all exceptions.
* **Finally, Block:** always runs, regardless of whether an exception occurred or not. It is typically used for cleanup operations.

Que. Understanding multiple exceptions and custom exceptions.

Answer:



To define a custom exception in Python, you need to create a new class that inherits from the built-in Exception class or one of its subclasses.

Example:

try:

result = 10 / 0

except ZeroDivisionError:

print("You cannot divide by zero!")

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

Answer:

Class:

A class is a user-defined data structure that allows us to bind attributes and their associated method together as a single unit.

Thus, class provides the facility of data encapsulation.

Syntax:

Class class\_name:

Attribute Declaration

Method Declaration

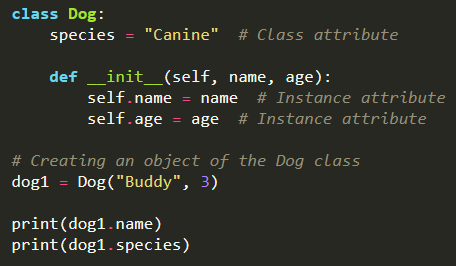
Object:

Once a class is defined, we can create an object of the class to access attributes and methods defined inside the class.

Thus, an Object is an instance of the class.

Syntax:

Object\_name = class\_name([Argument\_List])



Attribute:

 Attributes are the variables that belong to a class.

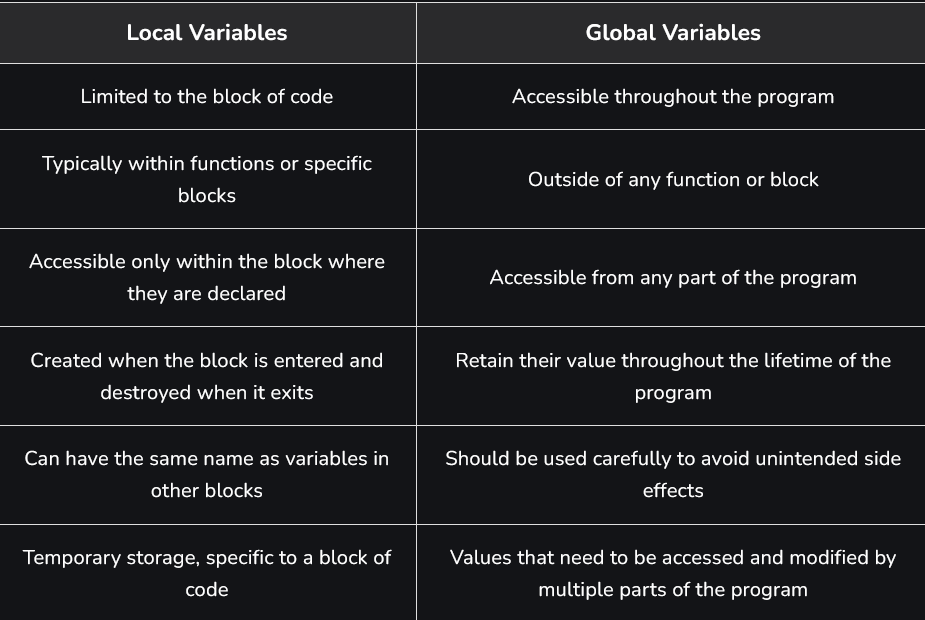
 Attributes are always public and can be accessed using the dot (.) operator.

Methods:

Methods are functions defined within a class that operate on the attributes of the class or object. They define the behaviour of the objects created from the class. Methods can take parameters and can return values. The first parameter of a method is typically self, which refers to the instance of the class.

Que. Difference between local and global variables.

Answer:



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

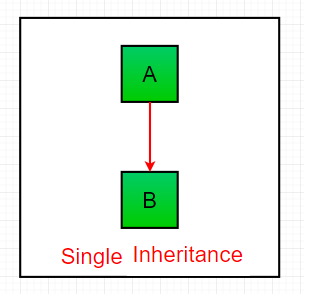
Answer:

The Most Important feature of inheritance is reusability.

### **Single Inheritance:**

The process of deriving a new class from an already existing class is known as Single Inheritance.

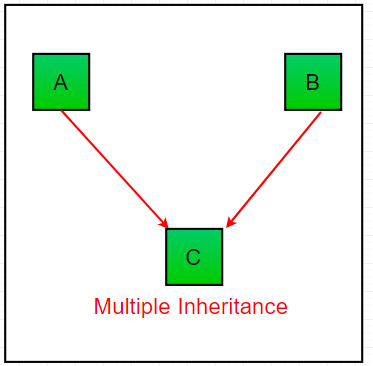
Thus, in Single inheritance, there is one base class and one derived class.



Multiple Inheritance:

The process of deriving a new class from more than one base class is known as multiple inheritance.

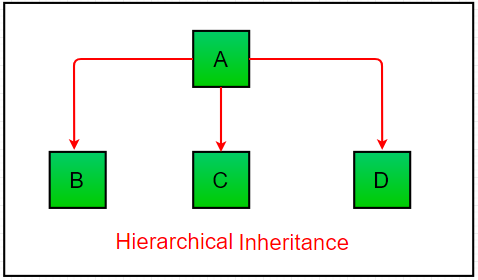
Thus, in multiple Inheritance, there is one derived class but more than one base class.



Hierarchical Inheritance:

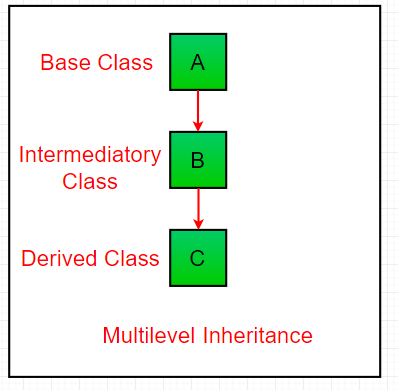
The process of deriving more than one class from a single base class is known as Hierarchical Inheritance.

Thus, in Hierarchical Inheritance, there is one base class but more than one derived class.



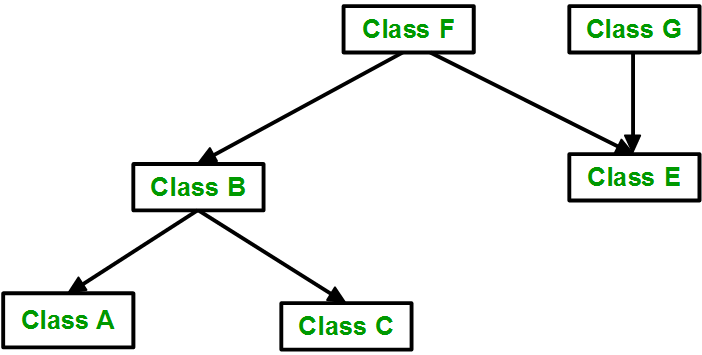
Multilevel Inheritance:

The process of deriving a new class from an already existing class and then again deriving a new class from the previously derived class is known as multilevel Inheritance.



Hybrid Inheritance:

A combination of more than one Inheritance is known as hybrid inheritance.



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

Answer:

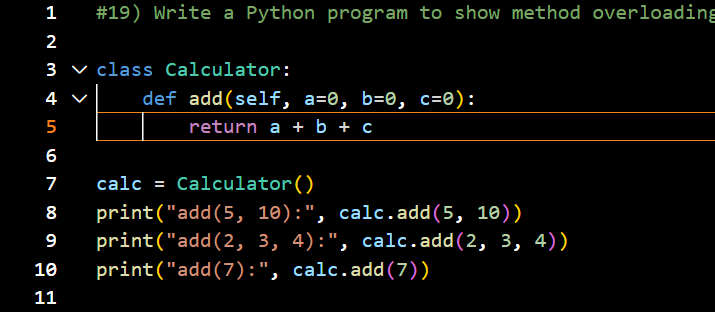
The super () function is used to refer to the parent class or superclass. It allows you to call methods defined in the superclass from the subclass, enabling you to extend and customize the functionality inherited from the parent class.

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

Parameters.

Answer:

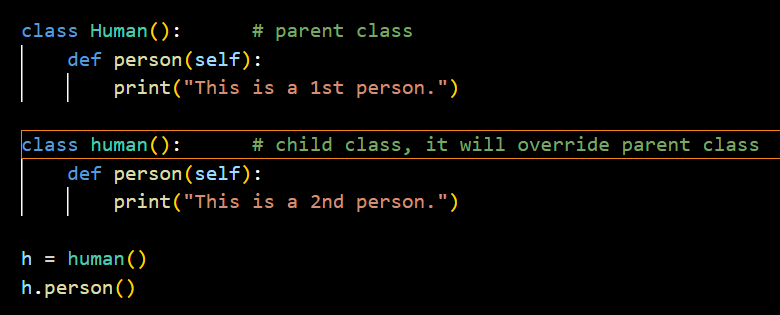
Two or more methods have the same name but different numbers of parameters or different types of parameters, or both. These methods are called overloaded methods and this is called method overloading.



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

Answer:

Method overriding is an ability of any object-oriented programming language that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, the same parameters or signature, and same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to **override** the method in the super-class.



Que. Introduction to SQLite3 and PyMySQL for database connectivity.

Answer:

SQLite3 is a lightweight, serverless database engine that allows you to create and manage databases directly from Python using the sqlite3 module. PyMySQL, on the other hand, is a pure Python MySQL client that enables connectivity to MySQL databases, allowing you to execute SQL queries and manage data effectively.

### SQLite3

* **Lightweight and Serverless**: SQLite3 is designed to be embedded within applications, making it ideal for small to medium-sized projects where a full database server is unnecessary.
* **Self-Contained**: The entire database is stored in a single file, which simplifies deployment and management.
* **Zero Configuration**: No setup is required; you can start using SQLite3 as soon as you import the sqlite3 module in Python.
* **ACID Compliance**: SQLite3 supports full ACID (Atomicity, Consistency, Isolation, Durability) transactions, ensuring data integrity.

### PyMySQL

* **Pure Python MySQL Client**: PyMySQL is a library that allows you to connect to MySQL databases using Python, providing a consistent API for executing SQL commands.

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

**matching.**

**Answer:**

The **re.search()** and **re.match()** both are functions of re module in python. These functions are very efficient and fast for searching in strings. The function searches for some substring in a string and returns a match object if found, else it returns none.

**re.match()** searches only from the beginning of the string and return match object if found. But if a match of substring is found somewhere in the middle of the string, it returns none.

**re.search()** searches for the whole string even if the string contains multi-lines and tries to find a match of the substring in all the lines of string.

Que. Difference between search and match.

Answer:

There is a difference between the use of both functions. Both return the first match of a substring found in the string, but **re.match()** searches only from the beginning of the string and return match object if found. But if a match of substring is found somewhere in the middle of the string, it returns none.   
While **re.search()** searches for the whole string even if the string contains multi-lines and tries to find a match of the substring in all the lines of string.