**Module 8) Advance Python Programming**

* **Printing on Screen**

1. Introduction to the print() function in Python

* The print() function outputs data to the standard output device (typically the console). It can accept multiple arguments, which are separated by spaces by default.
* print("Hello", "World") # Outputs: Hello World

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

* **f-strings (formatted string literals):** Introduced in Python 3.6, f-strings provide a concise and readable way to embed expressions inside string literals.

name = "Alice"

age = 30

print("My name is {name} and I am {age} years old.")

* **format() method:** Allows insertion of variables into strings using placeholders.

name = "Alice"

age = 30

print("My name is {} and I am {} years old.".format(name, age))

* Reading Data from Keyboard

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

* The input() function reads a line from input, returning it as a string.
* user\_input = input("Enter something: ")

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

* User input is read as a string; to use it as a number, it must be converted.

age = int(input("Enter your age: "))

height = float(input("Enter your height in meters: "))

* Opening and Closing Files

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

* 'r': Read mode (default).
* 'w': Write mode; creates a new file or truncates an existing one.
* 'a': Append mode; writes data at the end of the file.
* 'r+': Read and write mode; does not truncate the file.
* 'w+': Write and read mode; truncates the file.

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

* file = open("example.txt", "w")

3. Closing files using close().

* It's important to close a file after its operations are complete to free system resources.
* file.close()
* Reading and Writing Files

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

* read(): Reads the entire file.
* readline(): Reads one line at a time.
* readlines(): Reads all lines into a list.

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

* write(): Writes a string to the file.
* writelines(): Writes a list of strings to the file.
* Exception Handling

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

* Exceptions are errors detected during execution. In Python, exceptions can be handled using the try and except blocks. The finally block can be used to execute code regardless of whether an exception occurred or not.

try:

# Code that might raise an exception

except SomeException:

# Code that runs if the exception occurs

finally:

# Code that runs no matter what

2. Understanding multiple exceptions and custom exceptions.

* Multiple exceptions can be handled by specifying them as a tuple in an except block. Custom exceptions can be created by inheriting from the Exception class.

try:

# Code that might raise multiple exceptions

except (TypeError, ValueError) as e:

# Handle both exceptions

* Class and Object (OOP Concepts)

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

* A class is a blueprint for creating objects. It defines attributes (variables) and methods (functions).
* An object is an instance of a class.

class Car:

def \_\_init\_\_(self, brand, model):

self.brand = brand

self.model = model

mycar = Car("Mahindra", "Thar")

print(mycar.brand, mycar.model)

1. Difference between local and global variables.

* Local variables exist within a function and are deleted once the function exits.
* Global variables exist throughout the program.
* Inheritance

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

* **Single Inheritance:** A child class inherits from a single parent.
* **Multilevel Inheritance:** A child inherits from another child.
* **Multiple Inheritance:** A child inherits from multiple parents.
* **Hierarchical Inheritance:** Multiple child classes inherit from one parent.
* **Hybrid Inheritance:** Combination of the above types.