Python Interview Questions

1. What is list comprehension in Python. Give examples from the banking industry.

List comprehension in Python are a compact way to create a list. They can be thought of as syntactic sugar for transforming or filtering data for the purpose of creating new list.

**[expression for item in iterable if condition ]**

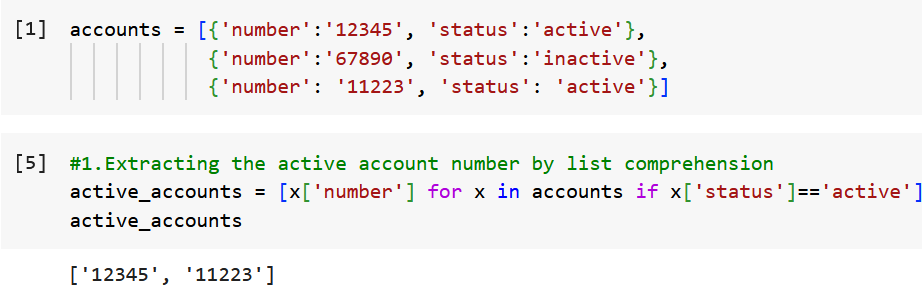
Where:

**Expression** is the current item in the iteration, but it could be transformed or used in some way.

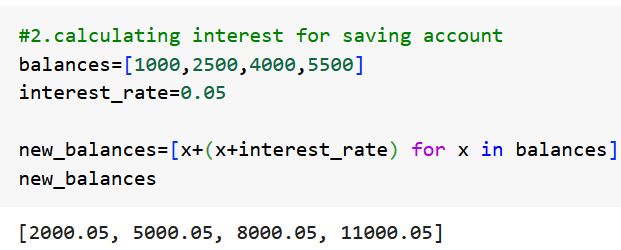
**Iterable** is a sequence (e.g. a list, tuple.string.etc) that can be iterated over.

**Condition** is like a filter that only through items for which the condition is true

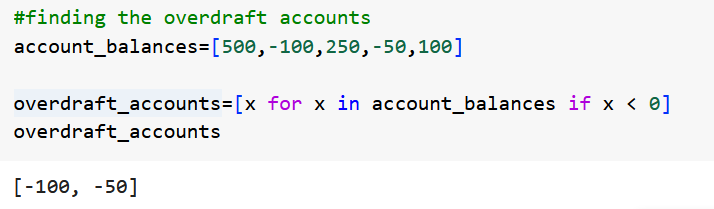
1.Extracting Active Account Numbers:



2.Calculating Interest for saving account

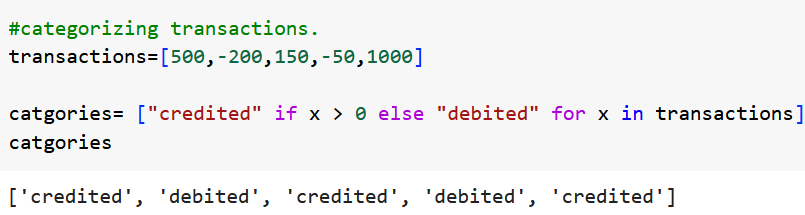


3.Finding overdrafts accounts – accounts having amount in -ve.

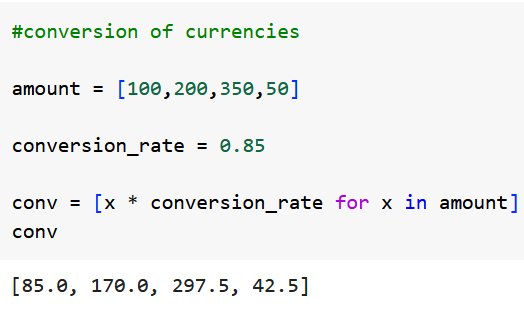


4.Categorizing transactions

Categorize transactions into debit and credit according to their amount



5.Converting currencies



List comprehension helps reducing amount of code written, making it more reliable and pythonic.

1. What is while loop, and where can we use that? Give 3 examples from the banking industry.

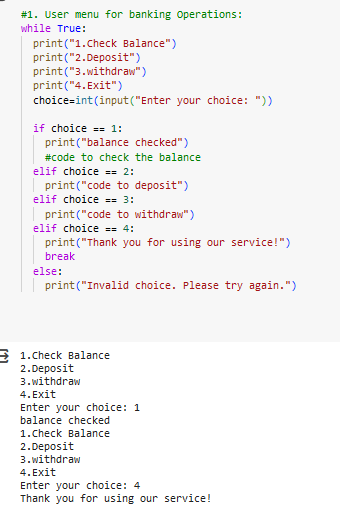
While Loop in Python is repeatedly executes a block of code as long as given condition is True. The loop terminates when the condition become False. The structure if while loop is:

While (condition):

#code to execute

The condition is evaluated before each iteration. If it’s true, the loop body is executed;if its False the loop terminates.

The while loop particularly useful when : The number of iterations are known in advance.



The loop proves useful for tasks that require repetition based on the specific conditions. In the banking industry , where operations often depend on various condition (e.g. fund clearance , user choice), the while loop becomes an invaluable tool.

1. What is for loop, and where can we use that? Give examples of the for loop from the banking industry.

A for loop in Python is used for iterating over a sequence (like list, tuple, dictionary, set or string) or other iterable objects. During each iteration, the loop variable takes on the value of the next item in the sequence.

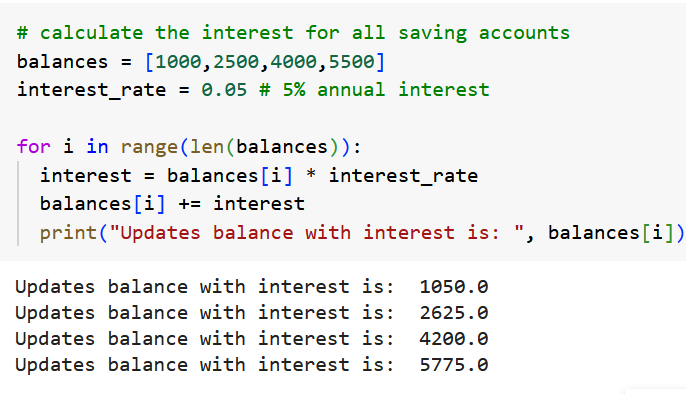
The general structure of a loop is:

Loop: for x in sequence:

#code to execute

Usage: The for loop is useful when: you know the number of iterations in advance, typically based on the length of a sequence,

You want to process each item in a sequence.

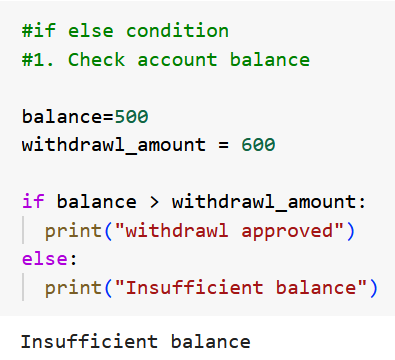
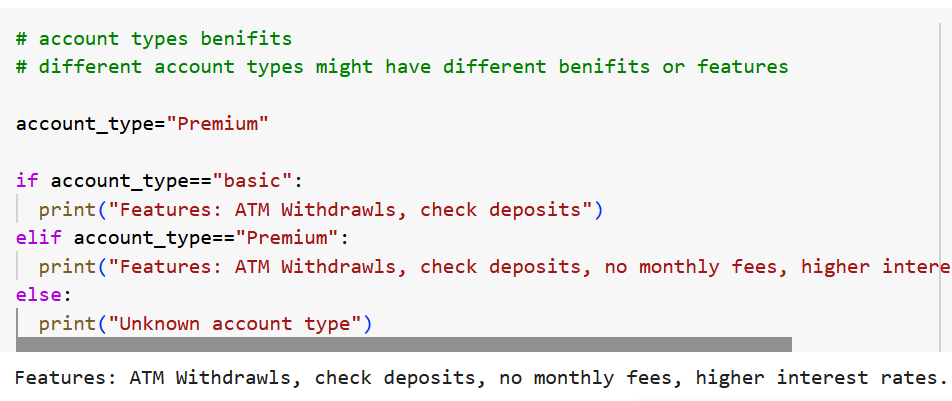
 

These examples showcase how the for loop can be employed in the banking sector to handle repetitive tasks that involves processing or iterating over sequences. The for loop provides a concise and clear way to handle such tasks, making the code more readable and maintainable.

1. What is the use of if and else condition in Python? Give banking example.

The if and else statement in python are used for the decision making. They allow us to execute certain blocks of code based on whether a condition is true or false.

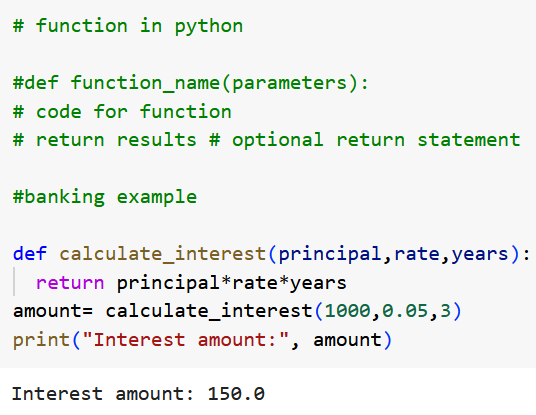
Usage: The if and else conditions are fundamentals for programming. They allow the program to make decisions, leading to dynamic and responsive behaviour based on different inputs or situations.

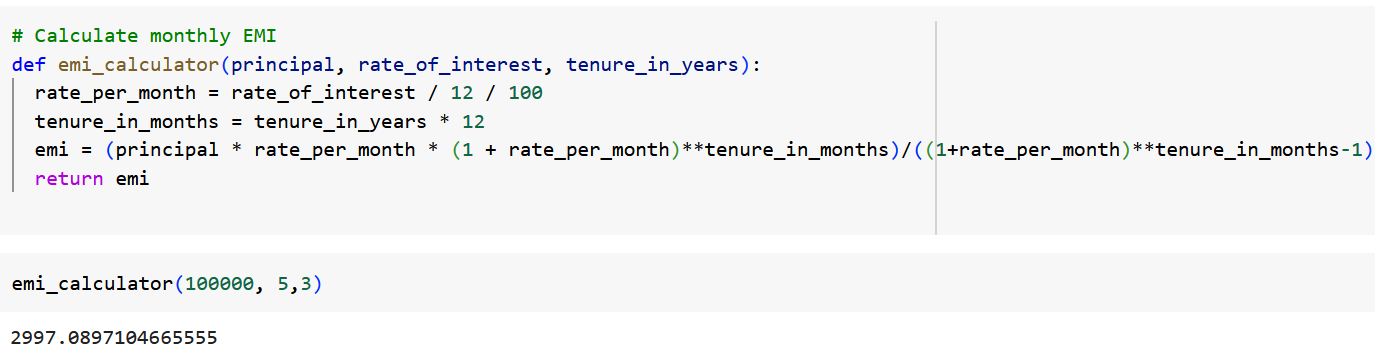
 

Decision making is crucial in any application and especially so in banking where different actionsor responces are needed based on the various fincancial situations or customer attributes.

1. How do you create functions in Python? Give some examples. Also, explain more about the \*args in python function.

In python, a function is a reusable block of code that performs a specific task. Function help in moduling the code, making it more organized, readable, and maintainable.

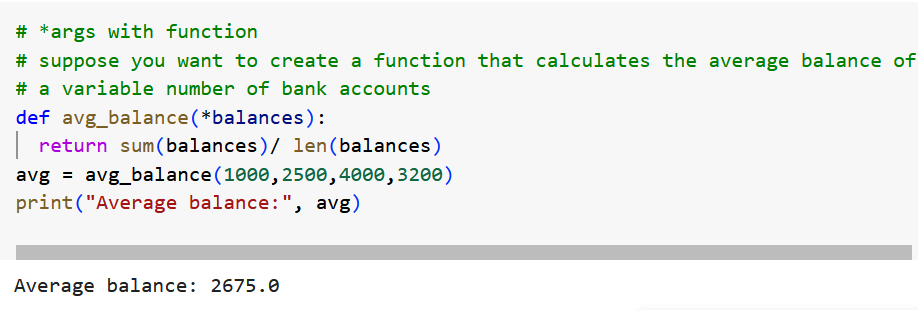




About \* args in Python Functions:

In python \*args is a way to pass a variable number of positional arguments to a function. It allows you to pass any number of positional arguments to the function, which are then accessible as a tuple within the function.

The syntax is to use an asterick(\*) before the parameter name(typically args, but the name itself is arbitrary; what matters is the asterick)

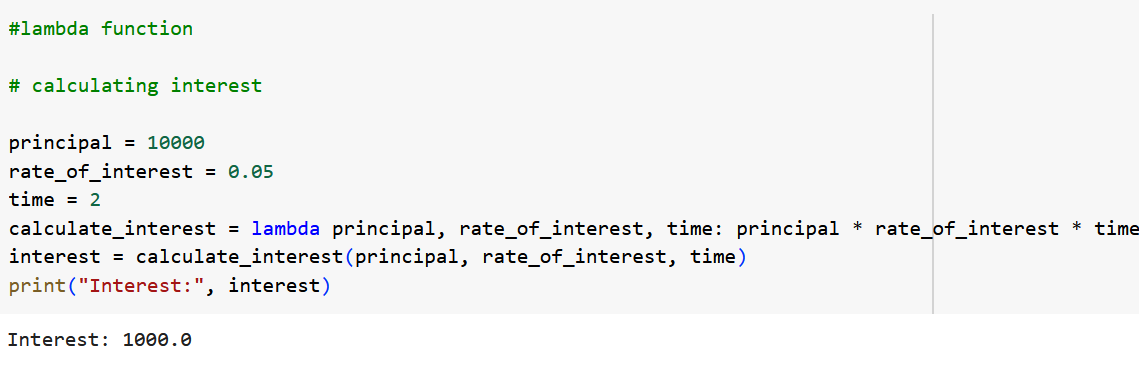


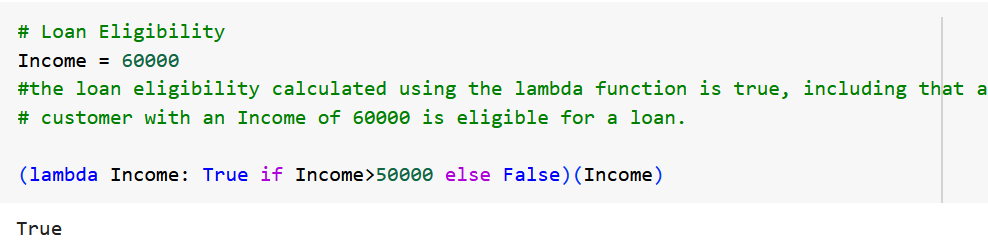
Using \*args provides flexibility and ensures that your functions can handle a varying number of inputs, which can be particularly useful in situations where the exact number of inputs might change or is not known in advance.

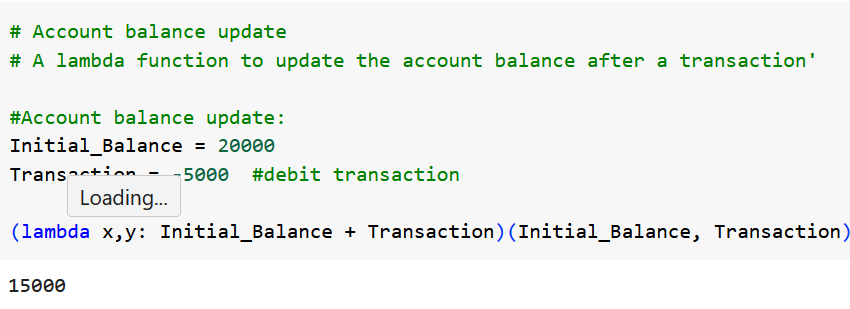
6. What is lambda function? Give examples to explain the same.

In Python, a lambda function is a small, anonymus function defined within the lambda keyword. Lambda functions can have any number of parameters but can have one expression. They are often used for short-term operations where a full function definition would feel overkill.

Lambda arguments: expression

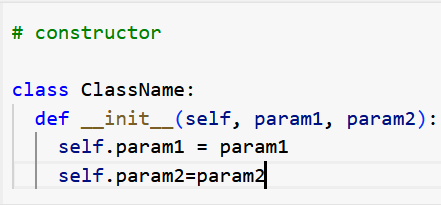






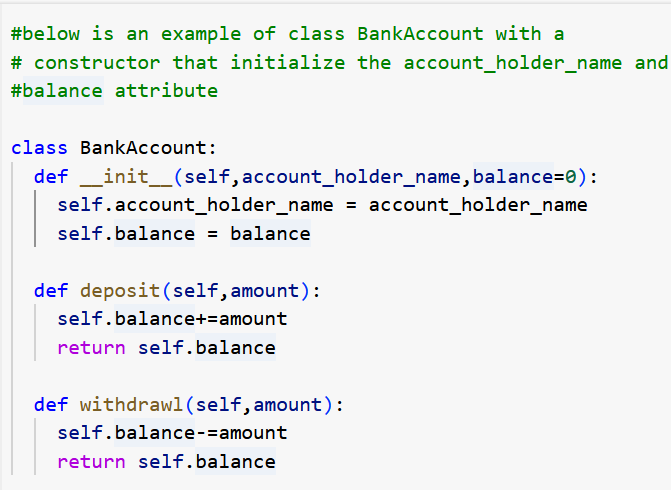
7. What is constructor in the OOPS in python? Give an example of the same.

In object-oriented programming, a constructor is a special method that is used to initialize the objects. In python, the constructor is defined using the **\_\_init\_\_** method. The **\_\_init\_\_** method is called automatically when a object is created from the class, and it allows the class to initialize attributes or perform any setup required when the object is being created.



Here the self represent the instance of the class and is used to access the variables and methods of the class.

Example:



Using the constructor.

Let’s create an object with the BankAccount and use its methods. The \_\_init\_\_ method will be called automatically when the object is created.

In the given example, we created an instance of BankAccount class named account with the account holder name as “John Doe” and an initial balance of 1000.

Here is the sequence of operations and the corresponding balances.

The initial balance, as set through the constructor is 1000.

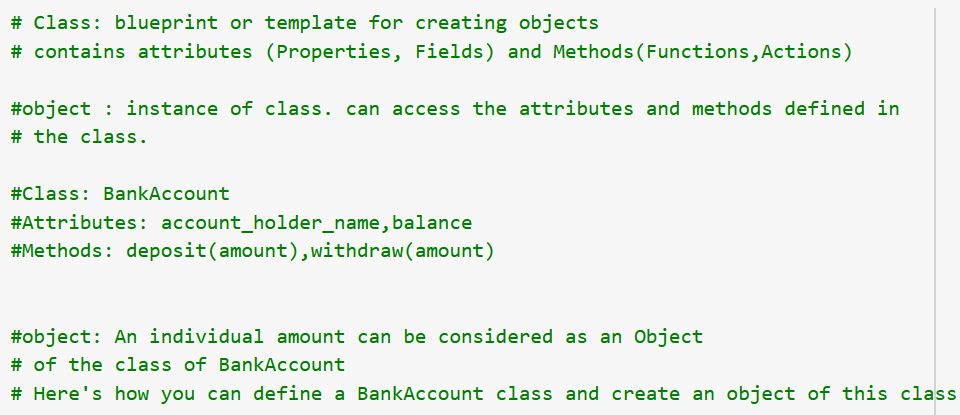
Deposit operation: An amount of 500 is deposited to account. After depositing 500 the new balance is 1500

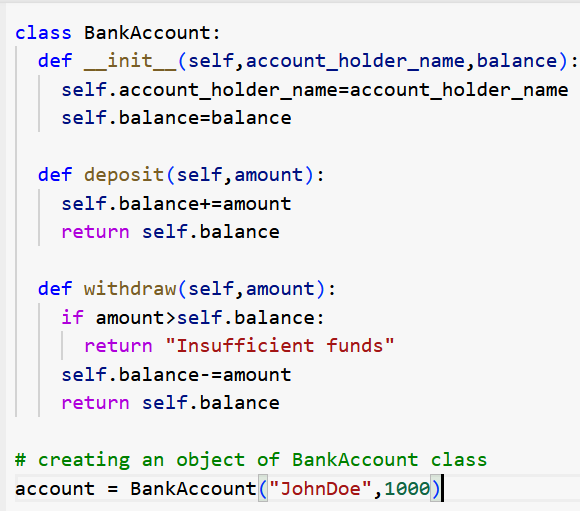
Withdraw operation: An amount of 300 is withdrawn from the account. After withdrawl the new balance will be 1200.

In this example the constructor (\_\_init\_\_) played a crucial role in initializing the attributes account\_holder\_name and balance of the object when it was created from the BannkAccount class.

8.What are the objects in OOPS. Give examples from the banking industry.

In OOPS a class is a blueprint for creating objects. A class defines the properties (also called attributes or fields) and methods (actions) that the objects created from the class will have. Objects are instances of classes containing data in the form of fields, often known as attributes, and code in the form of methods.





In this example:

BankAccount is a class that serves as a blueprint for bank accounts.

:account is an object of the class BankAccount class representing a specific bank account belonging to the “John Doe” with an initial balance of 1000.

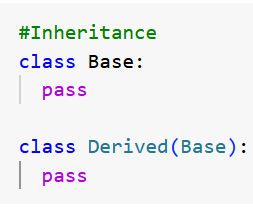
Account can access the deposit and withdraw methods to modify its balance attributes.

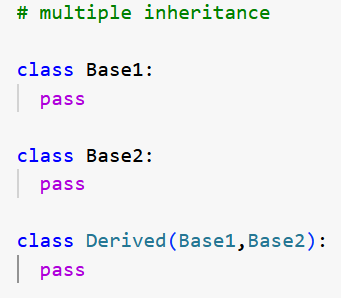
9.What are the different types of inheritances in Python OOPS languages?

Inheritance is a fundamental concept of OOPS, where a class (subclass/derived class) Inherits attributes and methods from another class (superclass/base class).

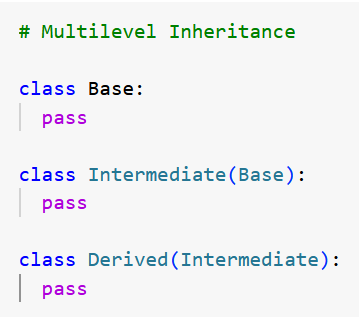
Python supports several types of inheritance:

1.Single Inheritance: In single inheritance, a class can inherit from only one superclass. This is the simplest form of inheritance.

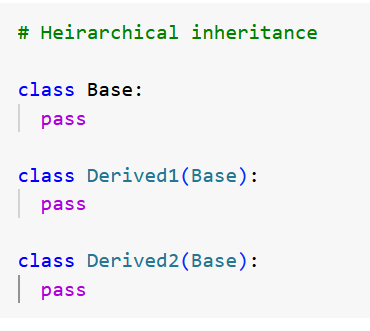


2.Multiple inheritance : In Multiple inheritance, a class can be derived from more than one base class. The derived class inherits attributes and methods from all base classes.

3.Multilevel Inheritance: In multilevel inheritance, a class is derived from a base class, which itself derived from another base class.

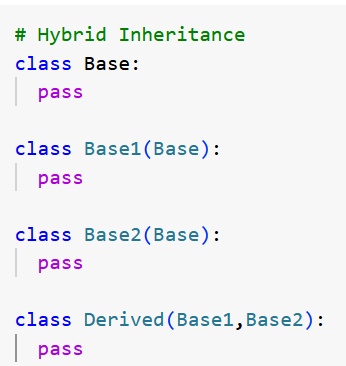


4.Heirarchical Inheritance: In hierarchical inheritance, one base class in inherited by more than one derived class.



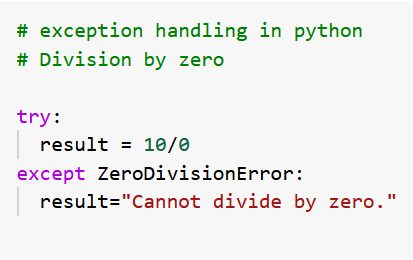
5.Hybrid Inheritance

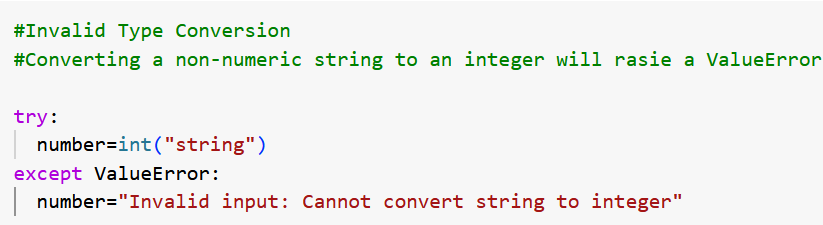
Hybrid inheritance is a combination of two or more type of inheritance. Due to complexity and ambiguity of multiple inheritance, Python uses the C3 linearization or C3 superclass linearization to order base classes in a specific sequence, known as the Method Resolution order.



10. What is exception handling in Python? Give examples

Exception handling in Python is a mechanism for gracefully responding to runtime errors or exceptions. An exception is an event that disrupts the normal flow of the program and is typically triggered when an error occurs. Python uses the try, except, else and finally blocks to handle the exceptions.







11. How do you open the file in python? Explain the different modes of opening a file in python.

To open a file in python, you can use ‘open()’ function takes two parameters: The file name/path and the mode in which you want to open the file. The basic syntax for opening the file is:

File = open(“filename”,”mode”)

Here “filename” represents the name of the file you want to open, and “mode” represents the mode in which you want to open the file.

There are several different modes for opening a file in Python, each serving a different purpose. Here are the most commonly used modes.

1. **“r” mode (Read mode)** : This is the default mode for opening the file. It allows you to read the contents of the file. If the file does not exist, it will raise a “FileNotFoundError”.

2. **“w” model (write mode)** : This mode is used for writing or creating a new file. If the file already exist, it will be truncated (i.e. all existing content will be deleted) . If the file does not exist, a new file will be created. If you try to read from the file in write mode, It will raise a “UnsupportedOperation” error.

3. **“a” mode (append mode)** : This mode is used for appending data to an existing file. If the file does not exist, a new file will be created. If you try to read from the file in append mode, it will raise a “UnsupportedOperation” error.

4. **“x” mode (Exclusive creation mode)** : This mode is used for writing or creating new file, but it will raise a “FileExistError” if the file already exists.

5. **“t” (Text mode)** : This is the default mode for opening the file in text format. It allows you to read or write the text data.

6. **“b” (Binary mode)** : this mode is used for opening the file in binary format. It allows you to read or write the binary data, such as images or audio files.

Additionally, you can combine modes by specifying them as string. For example, “r” represents read and write mode, allowing you to do both the read and modify operations to the file.

After opening the file, you can perform various operations on it, such as reading or writing the data. It is important to close the file after you are done using the “close()” method, to free up system resources. Alternatively, you can use the “with” statement, which automatically closes the file when you are done with it.