**ADITYA AMIN**

**ASSIGN : 08**

Q1. What are the two latest user-defined exception constraints in Python 3.X?

PEP 409 -- Unicode-based Emojis in Exception Messages: This enhancement allows user-defined exception classes to include Unicode-based emojis in their error messages. Prior to Python 3.7, emojis or other non-ASCII characters were not allowed in exception messages

PEP 526 -- Syntax for Variable Annotations: Although not specific to exceptions, PEP 526 introduced variable annotations, which can also be used in exception definitions. Variable annotations provide a way to specify the type of variables in Python code.

Q2. How are class-based exceptions that have been raised matched to handlers?

The exception is raised: When an exception is raised using the raise statement or by an error occurring within the program, Python starts searching for an exception handler.

Searching the call stack: Python examines the call stack, starting from the point where the exception was raised and moving up the stack. It looks for an appropriate except block that can handle the specific exception or its superclass.

Matching the exception: Python compares the raised exception to the exception types specified in each except block. It matches the exception based on the inheritance hierarchy, meaning that if the raised exception is an instance of a subclass, it can be caught by an except block for the superclass.

Handling the exception: When a matching except block is found, the code within that block is executed to handle the exception. If multiple except blocks match the raised exception, only the first matching block encountered during the traversal of the call stack is executed.

Propagation: If no matching except block is found in the call stack, the exception propagates to the next outer scope or the default exception handler if one exists. If no suitable handler is found, the program terminates, and an error message is displayed.

Q3. Describe two methods for attaching context information to exception artefacts.

Exception Arguments: When defining a custom exception class, you can include additional arguments to the exception constructor to capture and store context information. These arguments can be used to pass relevant data or objects that provide context about the exception.

Exception Attributes: Another method to attach context information to exceptions is by using attributes of the exception instance. You can define custom attributes on the exception object itself, which can hold additional data related to the exception.

Q4. Describe two methods for specifying the text of an exception object's error message.

Custom Exception Classes: You can define custom exception classes by subclassing built-in exception classes or the base Exception class. Within the custom exception class, you can override the \_\_init\_\_ method to accept arguments and customize the error message.

Formatting Strings: Another approach is to use formatted strings to construct the error message dynamically. This can be useful when you need to include variable values or other dynamic information in the error message.

Q5. Why do you no longer use string-based exceptions?

Loss of Information: String-based exceptions provide limited information about the nature of the exception. They lack the structured nature of class-based exceptions, which can carry additional attributes or data related to the exception.

Limited Error Handling: String-based exceptions make it challenging to selectively handle different types of exceptions.

Lack of Polymorphism: String-based exceptions lack the ability to leverage polymorphism, which is an essential concept in object-oriented programming. With class-based exceptions, you can define hierarchies of exception types and have different exception classes handle specific types of errors.

Code Readability and Maintenance: Class-based exceptions improve code readability and maintainability by providing self-documenting exception types. With string-based exceptions, it may be unclear what different exceptions represent without additional documentation or inspection of the exception message.