**ADITYA AMIN**

**ASSIGN : 06**

Q1. Describe three applications for exception processing.

Error Handling: Exception processing is primarily used for effective error handling in programs. By using try-except blocks, you can catch and handle specific types of exceptions that may arise during the execution of your code.

Input Validation: Exception processing can be employed to validate and handle invalid user input. For example, when accepting user input from a form or command-line interface, you can use try-except blocks to catch specific exceptions related to input validation, such as ValueError or TypeError.

Resource Management: Exception processing is crucial for proper resource management, such as file handling or network connections.

Q2. What happens if you don't do something extra to treat an exception?

If you don't handle or treat an exception explicitly in your code, it will propagate up the call stack until it reaches the default exception handler. This default handler will typically display an error message and terminate the program.

Q3. What are your options for recovering from an exception in your script?

Exception Handling with try-except: You can use a try-except block to catch and handle specific exceptions. Within the except block, you can implement code to handle the exception, such as logging an error message, displaying user-friendly feedback, or taking corrective actions

Exception Handling with Multiple except Blocks: If you anticipate multiple types of exceptions, you can use multiple except blocks to handle different exceptions separately.

Exception Handling with else Block: You can include an else block after the try-except block. The code within the else block will only execute if no exception is raised within the try block.

Q4. Describe two methods for triggering exceptions in your script.

Using the raise statement: You can use the raise statement to manually raise exceptions when certain conditions are met in your program. The raise statement takes an exception class or instance as an argument. For example, you can raise a ValueError exception when the user inputs an invalid value:

Using built-in functions or modules: Some built-in functions or modules in Python can raise exceptions when certain conditions are not met. For example, the open() function raises a FileNotFoundError exception if the specified file does not exist:

Q5. Identify two methods for specifying actions to be executed at termination time, regardless of whether or not an exception exists.

The finally block: The finally block is used to define a set of statements that will be executed regardless of whether an exception occurred or was caught. The code within the finally block will always run, providing a way to ensure certain operations are performed, such as closing files or releasing resources, regardless of the outcome.

The atexit module: The atexit module in Python provides a way to register functions to be called at program termination, regardless of whether an exception occurred or not. Functions registered using the atexit.register() function will be executed when the program exits normally, is interrupted by a signal, or an unhandled exception causes the program to terminate.