Q1. Describe three applications for exception processing.

Ans:  
Financial Transactions: Exception processing is crucial in the finance industry to detect and handle anomalies like fraudulent transactions, unusual account activities, or discrepancies in financial records.

Manufacturing Quality Control: In manufacturing, exception processing is used to identify and address defects or deviations from production standards, ensuring product quality and minimizing defects.

Healthcare Billing: Exception processing is employed in healthcare billing to flag and resolve billing discrepancies, insurance claim rejections, or coding errors, streamlining the revenue cycle management process.

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

Code will crash or you can say the excution will stop abruptly.  
Process Disruption: The normal workflow or process may be disrupted, leading to errors or unexpected outcomes.

Data Corruption: Unhandled exceptions can corrupt data, resulting in inaccuracies or data loss.

Security Risks: Vulnerabilities or security breaches may go unnoticed, posing significant risks to systems and data.

Customer Dissatisfaction: Failed exception handling can lead to poor user experiences, customer dissatisfaction, and reputational damage.

Q3. What are your options for recovering from an exception in your script?  
Ans:  
Retry: Attempt the operation again, assuming the issue was temporary or due to a transient problem.

Logging and Alerting: Log the exception details for debugging purposes and send alerts to notify administrators or users of the issue.

Fallback or Default Values: Provide default values or a fallback mechanism to continue processing when an exception occurs.

Exception Propagation: Allow the exception to propagate up the call stack, possibly to a higher-level error-handling mechanism.

Graceful Termination: If recovery is not possible, terminate the script gracefully, closing resources and preventing further damage.

User Interaction: Prompt the user for input or guidance on how to handle the exception in interactive scripts or applications.

Rollback: In database operations, perform a rollback to undo any changes made before the exception occurred.

Exception Handling Blocks: Use try-catch or try-except blocks to isolate and handle exceptions, allowing for specific recovery actions based on the exception type.

Q4. Describe two methods for triggering exceptions in your script.  
ans:  
Explicit Raise: You can explicitly raise an exception at a specific point in your code using a raise statement. For example:

Error-Prone Operations: Exceptions can also be triggered by error-prone operations, such as dividing by zero, accessing an undefined variable, or attempting to open a non-existent file. For example:

Q5. Identify two methods for specifying actions to be executed at termination time, regardless of whether or not an exception exists.  
Using the finally Block: In many programming languages, including Python and Java, you can use a finally block to specify code that should always be executed, whether or not an exception is raised. For example, in Python: