**Q1. Describe three applications for exception processing.**

Three applications for exception processing are:

* 1. Error Handling: Exception processing is used to handle and gracefully recover from errors or exceptional conditions in a program, preventing crashes and providing meaningful error messages.
* 2. Resource Management: It can be used for managing resources like files, network connections, or database connections by ensuring they are properly closed or released even in the presence of exceptions.
* 3. Input Validation: Exception processing can be used to validate user input or external data and raise exceptions when the data does not meet the required criteria, allowing for validation and error reporting.

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

* If we don’t handle exception will result in an error message and may lead to program termination, making it essential to implement proper exception handling to gracefully handle such situations**.**

**Q3. Describe two methods for triggering exceptions in your script.**

* 1. Explicitly Raise an Exception: You can use the `raise` statement to explicitly trigger an exception at a specific point in your code.

Example:

raise ValueError("This is a custom exception message.")

* 2. Invoke Built-in Functions: Many built-in functions and methods can raise exceptions under certain conditions. For instance, attempting to open a non-existent file using the `open()` function can trigger a `FileNotFoundError`:

Example:

with open("nonexistent\_file.txt", "r") as file: