1. What is the role of the 'else' block in a try-except statement? Provide an example scenario where it would be useful.
2. Can a try-except block be nested inside another try-except block? Explain with an example.
3. How can you create a custom exception class in Python? Provide an example that demonstrates its usage.
4. What are some common exceptions that are built-in to Python?
5. What is logging in Python, and why is it important in software development?
6. Explain the purpose of log levels in Python logging and provide examples of when each log level would be appropriate.
7. What are log formatters in Python logging, and how can you customise the log message format using formatters?
8. How can you set up logging to capture log messages from multiple modules or classes in a Python application?
9. What is the difference between the logging and print statements in Python? When should you use logging over print statements in a real-world application?
10. Write a Python program that logs a message to a file named "app.log" with the following requirements:
    - The log message should be "Hello, World!"
    - The log level should be set to "INFO."
    - The log file should append new log entries without overwriting previous ones.
11. Create a Python program that logs an error message to the console and a file named "errors.log" if an exception occurs during the program's execution. The error message should include the exception type and a timestamp.