Files, exceptional handling, logging and memory management

Assignment Questions

SKILLS

Files, exceptional handling, logging and memory management Questions

1. What is the difference between interpreted and compiled languages?
2. What is exception handling in Python?
3. What is the purpose of the finally block in exception handling?
4. What is logging in Python?
5. What is the significance of the __del__ method in Python?
6. What is the difference between import and from ... import in Python?
7. How can you handle multiple exceptions in Python?
8. What is the purpose of the with statement when handling files in Python?
9. What is the difference between multithreading and multiprocessing?
10. What are the advantages of using logging in a program?
11. What is memory management in Python?
12. What are the basic steps involved in exception handling in Python?
13. Why is memory management important in Python?
14. What is the role of try and except in exception handling?
15. How does Python's garbage collection system work?
16. What is the purpose of the else block in exception handling?
17. What are the common logging levels in Python?
18. What is the difference between os.fork() and multiprocessing in Python?
19. What is the importance of closing a file in Python?
20. What is the difference between file.read() and file.readline() in Python?
21. What is the logging module in Python used for?
22. What is the os module in Python used for in file handling?
23. What are the challenges associated with memory management in Python?
24. How do you raise an exception manually in Python?
25. Why is it important to use multithreading in certain applications?

Practical Questions

1. How can you open a file for writing in Python and write a string to it?
2. Write a Python program to read the contents of a file and print each line.
3. How would you handle a case where the file doesn't exist while trying to open it for reading?
4. Write a Python script that reads from one file and writes its content to another file.
5. How would you catch and handle division by zero error in Python?
6. Write a Python program that logs an error message to a log file when a division by zero exception occurs.
7. How do you log information at different levels (INFO, ERROR, WARNING) in Python using the logging module?
8. Write a program to handle a file opening error using exception handling.
9. How can you read a file line by line and store its content in a list in Python?
10. How can you append data to an existing file in Python?
11. Write a Python program that uses a try-except block to handle an error when attempting to access a dictionary key that doesn't exist.
12. Write a program that demonstrates using multiple except blocks to handle different types of exceptions.
13. How would you check if a file exists before attempting to read it in Python?
14. Write a program that uses the logging module to log both informational and error messages.
15. Write a Python program that prints the content of a file and handles the case when the file is empty.
16. Demonstrate how to use memory profiling to check the memory usage of a small program.
17. Write a Python program to create and write a list of numbers to a file, one number per line.
18. How would you implement a basic logging setup that logs to a file with rotation after 1MB?
19. Write a program that handles both IndexError and KeyError using a try-except block.
20. How would you open a file and read its contents using a context manager in Python?
21. Write a Python program that reads a file and prints the number of occurrences of a specific word.
22. How can you check if a file is empty before attempting to read its contents?
23. Write a Python program that writes to a log file when an error occurs during file handling.