29 Lecture - CS504

Important Mcqs

Q: In C++, which feature can be used for automatic resource management in file handling? a) try-catch block b) smart pointers c) dynamic memory allocation d) static variables **Solution: b) smart pointers** **Q: In Java, which statement is used for automatic resource management in file handling?** a) try-catch block b) finalize() method c) using statement d) try-with-resources **Solution: d) try-with-resources** **Q: What should you do before opening a file in C++ or Java?** a) Close any other open files. b) Check for file existence. c) Check the file size. d) Create a backup of the file. **Solution: b) Check for file existence.** **Q: Which file stream class in C++ provides buffered file input?** a) std::ifstream b) std::ofstream c) std::fstream d) std::stringstream **Solution: a) std::ifstream** **Q: In Java, which java.io class is commonly used for buffered file input?** a) BufferedReader b) FileWriter c) FileReader d) BufferedWriter **Solution: a) BufferedReader** **Q: What is the purpose of using relative paths in file handling?** a) To improve performance. b) To ensure file security. c) To handle large files efficiently. d) To avoid hardcoding absolute paths and improve portability. **Solution: d) To avoid hardcoding absolute paths and improve portability.** **Q: Which method is used for reading files line-by-line in Java?** a) readLine() b) read() c) readAllLines() d) readChar() **Solution: a) readLine()** **Q: How should files be closed after usage in both C++ and Java?** a) It is not necessary to close files manually. b) Use close() method in Java and delete keyword in C++. c) Use fclose() function in C++ and close() method in Java. d) Explicitly call close() method in Java and let C++ handle it automatically. **Solution: c) Use fclose() function in C++ and close() method in Java.** **Q: Which mode should be used for opening binary files in C++?** a) ios::out b) ios::binary c) ios::in d) ios::app **Solution: b) ios::binary** **Q: How should exceptions be handled during file handling?** a) Never use exceptions for file handling. b) Use try-catch blocks to handle exceptions gracefully. c) Use throws clause in Java and noexcept specifier in C++. d) Ignore exceptions to avoid program termination. **Solution: b) Use try-catch blocks to handle exceptions gracefully.**