

30 Lecture - CS504

Important Subjective

Q: What is the purpose of code layout in programming? **A:** Code layout aims to organize code in a structured and readable manner, making it easier to understand, maintain, and collaborate with other developers. **Q: How does consistent indentation improve code readability?** **A:** Consistent indentation visually represents code hierarchy, making it easier to identify nested blocks and understand the control flow of loops and conditionals. **Q: Why is it essential to use meaningful variable names in code?** **A:** Meaningful variable names convey the purpose and intent of the variables, improving code readability and reducing the need for excessive comments. **Q: How can comments improve code documentation?** **A:** Comments provide explanations and context within code, helping developers understand complex logic, design decisions, and usage of functions or classes. **Q: What are the different types of comments used in Java and C++?** **A:** Both Java and C++ support single-line comments (`//`) and multi-line comments (`/* ... */`). **Q: Why should developers avoid excessive comments in code?** **A:** Excessive comments can clutter the code and make it harder to maintain, especially if the comments are redundant or not kept up-to-date with code changes. **Q: How do multi-line comments help in documenting code?** **A:** Multi-line comments are used to provide detailed explanations and documentation for functions, classes, or sections of code that require more context. **Q: How can layouts and comments facilitate code collaboration in a team?** **A:** Consistent layouts and meaningful comments promote code understanding among team members, fostering collaboration and efficient problem-solving. **Q: In C++, how does the RAII principle relate to proper file layout?** **A:** The RAII (Resource Acquisition Is Initialization) principle encourages using smart pointers and classes to manage resources like files, ensuring proper file handling and closure. **Q: How do relative file paths contribute to code portability?** **A:** Relative file paths allow the code to access files consistently across different platforms, avoiding platform-specific issues with absolute paths.