

12 Lecture - CS410

Important Subjective

1. What is a window class in the context of graphical user interface programming?

Answer: A window class is a template or blueprint that defines the attributes and behavior of a window in a graphical user interface application. It includes information such as the window procedure, style, icon, cursor, background color, and class name.

2. Explain the purpose of the window procedure in a window class.

Answer: The window procedure is a function that processes messages sent to a window. It handles events such as mouse clicks, keyboard input, and system messages, allowing the window to respond appropriately to user interactions and system events.

3. How do you register a window class in Windows API?

Answer: To register a window class in Windows API, you need to call the RegisterClassEx function, passing a pointer to a WNDCLASSEX structure that contains the window class's attributes.

4. What is the significance of the "hInstance" parameter in the RegisterClassEx function?

Answer: The "hInstance" parameter represents the handle to the instance of the application (the executable file). It identifies which program's window class is being registered, allowing Windows to associate the class with the correct application.

5. Can a window class be shared across multiple windows in an application?

Answer: Yes, a window class can be shared across multiple windows in an application. By using the same window class name during window creation, multiple windows can share the same template and behavior.

6. What happens if you attempt to create a window with an unregistered window class?

Answer: If you try to create a window with an unregistered window class, the window creation process will fail. Before creating a window, the associated window class must be registered using the RegisterClassEx function.

7. How do you handle messages in the window procedure?

Answer: In the window procedure, messages are typically processed using a switch statement based on the message identifier (WM_XXX). The appropriate code is executed for each specific message to handle user input and system events.

8. What is the purpose of the "lpParam" parameter in the CreateWindowEx function?

Answer: The "lpParam" parameter in the CreateWindowEx function allows you to pass additional user-defined data to the window procedure. It provides a way to communicate custom information specific to the window being created.

9. Can you change the window class attributes after registering the window class?

Answer: No, you cannot change the window class attributes after registering the window class. The attributes are set during registration and remain constant for all windows created using that class.

10. How do you free resources associated with a window class when it is no longer needed?

Answer: To free resources associated with a window class, you should call the UnregisterClass function when all windows created with that class are destroyed. This ensures that the resources are properly released and avoids memory leaks.