DIGITAL CLOCK USING C++

This C++ code is a simple program to create a digital clock with basic user input for setting the initial time (hours, minutes, and seconds). The program will display the clock on the console and update it every second until the user terminates the program manually.

Let's go through the code step by step to understand it:

#include<iostream>: This line includes the input-output stream library, allowing the program to perform input and output operations.

#include<windows.h>: This line includes the Windows API library, which provides access to functions like Sleep that allow the program to pause execution for a specified amount of time.

using namespace std;: This line avoids having to write **std**:: before standard library functions.

The **main()** function: The program starts executing from here.

Variable declarations:

int h, m, s, a, err;: These variables are used to store the user input for hours (h), minutes (m), seconds (s), and some auxiliary flags (a and err).

err = a = 0;: Initializing the err and a variables to 0.

First while loop **(while (err == 0)):**

This loop is designed to take user input for the initial time (hours, minutes, seconds) and validate whether the input is within the valid range (hours < 24, minutes < 60, seconds < 60).

If the input is valid, err is incremented (making the loop exit).

If the input is not valid, the screen is cleared using **system("cls"),** and the user is prompted to enter the time again.

Second while loop **(while (a == 0)):**

This loop is the main clock display loop, which will keep updating the time every second and display it on the console.

The loop starts by clearing the screen using **system("cls").**

The current time (stored in h, m, and s variables) is displayed in the format "hh:mm:ss".

The **Sleep(1000)** function is used to pause execution for 1000 milliseconds (1second).

The seconds (s) are incremented, and if they exceed 59, they are reset to 0, and the minutes (m) are incremented.

Similarly, if minutes exceed 59, they are reset to 0, and the hours (h) are incremented.

If hours exceed 24, they are reset to 0, creating a simple 24-hour digital clock.

The loop continues indefinitely, updating the time every second until the program is terminated manually (by pressing Ctrl+C or closing the console window).

return 0;: This statement indicates that the program has executed successfully, and the **main() function returns 0.**

To summarize, this program creates a basic digital clock that allows the user to input the initial time and then continuously updates and displays the time every second.