Students Handout

Introduction to C# Workshop 1 - Admin Profile Management

Introduction:

This workshop aims to introduce you to the basics of C# programming through a simple program that manages admin profiles. In this session, you'll learn about user input, data storage, and displaying information using the C# programming language.

Program Overview:

The program is designed to perform two main functions: registering an admin and displaying admin details. Let’s explore the functionality step by step:

Registering an Admin (Option A):

* When prompted, choose the option 'A' to register as an admin.
* Input the admin's name, username, email, and phone number as directed.
* The program stores this information in memory for later access.

Displaying Admin Details (Option D):

* To view the admin details, select option 'D'.
* The program retrieves and displays the previously entered admin information.

Program Flow:

* The program starts by displaying a welcome message to the Admin section.
* It presents a menu with options to Register as Admin (A) or Display Admin Details (D).
* Depending on the selected option, it performs the respective action as explained above.
* After each action, it prompts whether to continue (Y/N).

Understanding the Code:

The main aim of this code is to demonstrate the use of conditional statements (switch) and a loop (do-while) in a C# program. The program allows users to register as an admin and display admin details based on their choices. The switch statement is used to execute different actions depending on the user's input, and the do-while loop ensures that the program continues running until the user decides to exit.

* Namespaces: Importing necessary libraries like System for basic functionalities.
* AdminProfile Struct: Defines the structure for storing admin information (name, username, email, phone number).
* Main Method: The entry point of the program where the code execution begins.
* Switch Statement: Determines the action based on user input (A or D).
* Do-While Loop: Executes the program until the user chooses to exit (by inputting 'N' or 'n').
* Console Inputs and Outputs: Using Console.ReadLine() to get user input and Console.WriteLine() to display output messages.

Important Notes:

* Ensure to input data accurately as the program stores and displays admin information based on user inputs.
* This program currently manages a single admin profile. Enhancements can be made to handle multiple admins by using arrays or collections.

Conclusion:

This program is a basic demonstration of how user input and data storage work in C#. Explore and experiment with it to deepen your understanding of programming concepts.