**Simple Delegate for Greeting Messages**

**Problem Statement:** Simple Delegate for Greeting Messages

**Objective**: Create a console application that dynamically selects and displays different types of greeting messages (formal or informal) based on user input, using a delegate to reference the appropriate greeting method.

**Description**:

The application should allow users to choose between a formal greeting (e.g., " Pleasure to meet you!") or an informal greeting (e.g., "Hey, what's up?") for a given name. A delegate will be used to reference the greeting method, enabling dynamic selection of the method to invoke at runtime. The program should:

- Prompt the user to select the type of greeting (formal or informal).

- Accept a name as input (hardcoded or user-provided).

- Use a delegate to call the appropriate greeting method based on the user's choice.

- Display the selected greeting message.

**Requirements**:

1. Define a delegate type that matches the signature of methods that take a string (name) and return void.

2. Implement at least two greeting methods (formal and informal) that match the delegate's signature.

3. Allow the user to input their choice of greeting type (e.g., "formal" or "informal").

4. Assign the appropriate method to the delegate based on the user's input.

5. Invoke the delegate to display the greeting for a specified name.

6. Handle invalid input gracefully by defaulting to one of the greeting methods.

**Expected Input:**

- User input for greeting type: "formal" or "informal" (case-insensitive).

- A name (e.g., "Alice") to include in the greeting.

**Expected Output:**

- If the user selects "formal": `Dear Alice, it is a pleasure to meet you.`

- If the user selects "informal": `Hey Alice, what's up?`

- If the input is invalid: Default to one of the greetings (e.g., informal).

**Constraints**:

- The application should be simple and console-based.

- The delegate must be used to invoke the greeting method.

- Input validation is optional but recommended for robustness.

**Learning Goals:**

- Understand the concept of delegates as function pointers in C#.

- Learn how to define and use a delegate to dynamically select and invoke methods.

- Gain experience with basic console input/output and conditional logic in C#.