C Program for Temperature Conversion

```
#include <stdio.h>

// Function to convert Celsius to Fahrenheit
float celsiusToFahrenheit(float celsius) {
    return (celsius * 9.0 / 5.0) + 32;
}

// Function to convert Fahrenheit to Celsius
float fahrenheitToCelsius(float fahrenheit) {
    return (fahrenheit - 32) * 5.0 / 9.0;
}

int main() {
    int choice;
    float temperature;

    printf("Temperature Conversion Program\n");
    printf("1. Celsius to Fahrenheit\n");
    printf("2. Fahrenheit to Celsius\n");
    printf("Enter your choice (1 or 2): ");
    scanf("%d", &choice);

    // Input the temperature based on the user's choice
    switch(choice) {
        case 1:
            printf("Enter temperature in Celsius: ");
            scanf("%f", &temperature);
            printf("%.2f Celsius = %.2f Fahrenheit\n", temperature,
celsiusToFahrenheit(temperature));
```

```
            break;
        case 2:
            printf("Enter temperature in Fahrenheit: ");
            scanf("%f", &temperature);
            printf("%.2f Fahrenheit = %.2f Celsius\n", temperature,
fahrenheitToCelsius(temperature));
            break;
        default:
            printf("Invalid choice! Please choose either 1 or 2.\n");
            break;
    }

    return 0;
}
```

Logic of the Program:

1. **Function Definitions**:
    - The program defines two functions:
        - `celsiusToFahrenheit(float celsius)` converts a given temperature in Celsius to Fahrenheit using the formula:
          F=95×C+32F = \frac{9}{5} \times C + 32F=59×C+32
        - `fahrenheitToCelsius(float fahrenheit)` converts a given temperature in Fahrenheit to Celsius using the formula:
          C=59×(F−32)C = \frac{5}{9} \times (F - 32)C=95×(F−32)
2. **User Input**:
    - The user is prompted to choose whether they want to convert from Celsius to Fahrenheit or from Fahrenheit to Celsius.

    - Based on the choice, the program asks the user to input the temperature in the appropriate unit (Celsius or Fahrenheit).
3. **Conversion Process**:
    - If the user chooses option 1 (Celsius to Fahrenheit), the program converts the input Celsius value to Fahrenheit using the `celsiusToFahrenheit()` function.
    - If the user chooses option 2 (Fahrenheit to Celsius), the program converts the input Fahrenheit value to Celsius using the `fahrenheitToCelsius()` function.
4. **Display the Result**:
    - The program displays the result of the conversion with two decimal points for precision.

Sample Output:

Sample Output for Celsius to Fahrenheit conversion:
Temperature Conversion Program
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter your choice (1 or 2): 1
Enter temperature in Celsius: 25
25.00 Celsius = 77.00 Fahrenheit

Sample Output for Fahrenheit to Celsius conversion:
Temperature Conversion Program
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter your choice (1 or 2): 2
Enter temperature in Fahrenheit: 77
77.00 Fahrenheit = 25.00 Celsius

Explanation of the Code:

1. **Header File Inclusion**:
    - `#include <stdio.h>` is used to include the Standard Input/Output header file for using functions like `printf()` and `scanf()`.
2. **Functions**:
    - The `celsiusToFahrenheit` and `fahrenheitToCelsius` functions perform the necessary conversions using the respective formulas.
3. **Main Program**:
    - The program begins by printing a menu for the user to choose which conversion they want to perform.
    - It then takes user input for the choice and the temperature value, performs the conversion using the appropriate function, and prints the result.
4. **Switch Statement**:
    - A `switch` statement is used to handle the user's choice for the conversion type. If the user selects option 1, the program converts Celsius to Fahrenheit. If option 2 is selected, the program converts Fahrenheit to Celsius.
5. **Error Handling**:
    - If the user enters an invalid choice (i.e., something other than 1 or 2), an error message is displayed.

Conclusion:

This program offers a simple but efficient way to convert temperatures between Celsius and Fahrenheit. It uses functions for modularity, making the code more organized and reusable. The program's flow ensures that users can easily interact with it and get the desired conversion result.