Temperature Conversion System Using System Exceptions

Title

Temperature Conversion Utility with Exception Handling

Question No : 1 / 1

Problem Statement

Write a C# console application that reads a temperature value and a conversion type from the user.
The conversion type indicates whether to convert the temperature from **Fahrenheit to Celsius** or from **Celsius to Fahrenheit**.

The program should output the converted temperature rounded to **two decimal places**.

If the user provides **invalid input** (non-numeric temperature or invalid conversion type), the program should handle the error gracefully and display an appropriate error message.

Use **SystemException** to catch the exception message.
Write the solution within the **Program.cs** file.

Requirements

1. Input

- A numeric value representing the temperature.
- A string representing the conversion type:
    - **F** → Fahrenheit to Celsius
    - **C** → Celsius to Fahrenheit

2. Output

- Converted temperature rounded to two decimal places.
- Error message for invalid input.

3. Conversion Formula

- Fahrenheit to Celsius
  ```
  celsius = (temperature - 32) * 5 / 9
  ```
- Celsius to Fahrenheit
  ```
  fahrenheit = (temperature * 9 / 5) + 32
  ```

Input Format

- First line: Temperature value of type **double**
- Second line: Conversion type (**C** or **F**) of type **string**

Output Format

- If input is **F**
- `Temperature in Celsius: {value}`
- If input is **C**
- `Temperature in Fahrenheit: {value}`
- If conversion type is invalid
- `Invalid conversion type. Please enter 'F' or 'C'.`
- If temperature input is invalid
- `Error: Invalid input provided.`
- `Exception Message: {errorMessage}`

Test Cases

Test Case ID	Input	Expected Output
TC01	`100, F`	Temperature in Celsius: 37.78
TC02	`30, C`	Temperature in Fahrenheit: 86.00
TC03	`abc`	Error: Invalid input provided. Exception Message: Input string was not in a correct format.
TC04	`10, g`	Invalid conversion type. Please enter 'F' or 'C'.
TC05	`-40, F`	Temperature in Celsius: -40.00

Sample Input 1

```
100
F
```

Sample Output 1

```
Temperature in Celsius: 37.78
```

Sample Input 2

```
30
C
```

Sample Output 2

```
Temperature in Fahrenheit: 86.00
```

Sample Input 3

```
abc
```

Sample Output 3

```
Error: Invalid input provided.
Exception Message: Input string was not in a correct format.
```

Sample Input 4

```
10
g
```

Sample Output 4

```
Invalid conversion type. Please enter 'F' or 'C'.
```

C# Solution (Program.cs)

```csharp
using System;

namespace TemperatureNamespace
{
  public class TemperatureConversion
  {
    public static void Demo()
    {
      try
      {

        double temperature = Convert.ToDouble(Console.ReadLine());
        string conversionType = Convert.ToString(Console.ReadLine());
        if (conversionType.ToUpper() == "F")
        {
          double convertedTemperature = (temperature - 32) * 5 / 9;
          Console.WriteLine("Temperature in Fahrenheit: {0:F2}", convertedTemperature);
        }
        else if (conversionType.ToUpper() == "C")
        {
          double convertedTemperature = temperature * 9 / 5 + 32;
          Console.WriteLine("Temperature in Celsius: {0:F2}", convertedTemperature);
        }
        else
        {
```

```
        throw new TypeException("Invalid conversion type.Please enter 'F' or 'C'.");

      }

    }

    catch (FormatException e)

    {

      Console.WriteLine("Error: Invalid input provided.");

      Console.WriteLine($"Exception Message: {e.Message}");

    }

    catch (TypeException e)

    {

      Console.WriteLine(e.Message);

    }

  }

 }

}
```

Title

Cricket Match Score Tracking System with Exception Handling

Question No : 1 / 1

Problem Statement

You are required to develop a **C# console application** that simulates a simple **cricket match score tracking system**.

The application allows the user to input scores of players, validates the scores, and calculates the **total score of the cricket team**.

The application should handle various exceptions that may arise during the **input and processing of scores**.

Write the solution within the **Program.cs** file.

Requirements

All classes must be public.

---

1. Class: CricketMatch

Properties

- int[] playerScores
  → An array to store scores of **up to 5 players**

- int currentIndex
  → Tracks the number of scores added

Methods

- void AddPlayerScore(int score)

  o Adds a player's score to the array

  o Throws:

    ▪ ArgumentException
      → If score is **less than 0 or greater than 50**
      → Message:

    ▪ Invalid score. Score must be between 0 and 50.

    ▪ InvalidOperationException
      → If more than **5 scores** are added
      → Message:

    ▪ Cannot add more than 5 player scores.

- int CalculateTotalScore()

  o Calculates and returns the **total team score**

2. Class: Program

Main Method Responsibilities

- Reads space-separated player scores from console input

- Adds each score using AddPlayerScore

- Displays the total score using CalculateTotalScore

- Handles the following exceptions:

  - ArgumentException

  - InvalidOperationException

Exception Handling Rules

- For **more than 5 scores**

- throw new InvalidOperationException("Cannot add more than 5 player scores.");

- For **invalid score values**

- throw new ArgumentException("Invalid score. Score must be between 0 and 50.");

Input Format

- The first line of input should be **space-separated integers** representing player scores.

Output Format

- If number of players ≤ 5 and all scores are valid:

- Total score of the cricket team: {calculatedTotalScore}

- If number of scores > 5:

- Error: Cannot add more than 5 player scores.

- If any score is invalid:

- Error: Invalid score. Score must be between 0 and 50.

Test Cases

Test Case ID	Input	Expected Output
TC01	1 2 3 4	Total score of the cricket team: 10
TC02	1 2 3 4 5	Total score of the cricket team: 15
TC03	1 2 3 4 5 6	Error: Cannot add more than 5 player scores.
TC04	40 -1 20	Error: Invalid score. Score must be between 0 and 50.
TC05	60 10	Error: Invalid score. Score must be between 0 and 50.

Sample Input 1

1 2 3 4

Sample Output 1

Total score of the cricket team: 10

Sample Input 2

1 2 3 4 5 6

Sample Output 2

Error: Cannot add more than 5 player scores.

Sample Input 3

40 -1 24

Sample Output 3

Error: Invalid score. Score must be between 0 and 50.

```csharp
using System;

namespace CricketMatchScoreNamespace
{
 public class CricketMatch
 {

   int[] playerScores;
   int currentIndex;

   public CricketMatch()
   {
    playerScores=new int[5];
    currentIndex =0;
   }
   public void AddPlayerScore(int score)
   {
   if (score < 0 || score > 50)
   {
     throw new ArgumentException("Invalid score. Score must be between 0 and 50.");
   }
    else if (currentIndex > 5)
   {
     throw new InvalidOperationException("Cannot add more than 5 player scores.");
```

```
        }
        else
        {
            playerScores[currentIndex] = score;
            currentIndex++;
        }
    }
    public int CalculateTotalScore()
    {
        int totalScore = 0;
        for (int i = 0; i < currentIndex; i++)
        {
            totalScore += playerScore[i];
        }
        return totalScore;
    }
}
public class Program
{
    public static void Main()
    {

        CricketMatch match = new CricketMatch();
        try
        {
```

```csharp
            string stringScore = Console.ReadLine();

            string[] playerScore = stringScore.Split(' ');

            for (int i = 0; i < playerScore.Length; i++)

            {

                match.AddPlayerScore(Convert.ToInt32(playerScore[i]));

            }

        }

        catch (ArgumentException e)

        {

            System.Console.WriteLine(e.Message);

        }

        catch (InvalidOperationException e)

        {

            System.Console.WriteLine(e.Message);

        }

        catch(FormatException e)

        {

            System.Console.WriteLine(e.Message);

        }

        finally

        {

            int calculatedTotalScore= match.CalculateTotalScore();

            System.Console.WriteLine($"Total score of the cricket team: {calculatedTotalScore}");

        }
```

}

}

}