12214994_SaurabhRana

Question No : 1 / 1

Multi-Unit Converter System

Write a C# program that converts values between different units within the same category (length, weight, or temperature). The program should be able to convert between specific units as well as convert to a default unit for each category.

The user should be able to specify:

- The value to convert
- The unit to convert from
- Optionally, the unit to convert to

If the target unit is not specified, the program should convert to the default unit for that category.

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

Requirements

The program should support the following conversions:

Length

- Meters to Kilometers and vice versa
- Miles to Feet and vice versa

Weight

- Grams to Kilograms and vice versa
- Pounds to Ounces and vice versa

Temperature

- Celsius to Fahrenheit and vice versa
- Celsius to Kelvin and vice versa

The program should output the converted value along with the unit.

Method Overview

UnitConverter Class

- Create a public class called **UnitConverter**

Convert Method (Overloaded)

- `public double Convert(double value, string fromUnit, string toUnit)`
  - Converts value from `fromUnit` to `toUnit`
  - Includes specific conversion logic for each unit type
- `public double Convert(double value, string fromUnit)`
  - Converts value from `fromUnit` to the default unit for its category
  - Calls the specific Convert method with the appropriate default unit

---

Program Class

Main Method

- Reads input values for:
  - Type of conversion
  - Value to convert
  - From unit
  - Optionally, to unit
- Calls the appropriate Convert method from the UnitConverter class
- Prints the conversion result

GetDefaultUnit(string type)

Purpose

Returns the default unit for a given type of conversion.

Operation

Uses a switch statement to return the default unit based on the type of conversion:

- Length → `"meters"`
- Weight → `"kilograms"`
- Temperature → `"celsius"`

Formulas for Conversions

1. Length Conversions

- Meters to Kilometers:
  `1 meter = 0.001 kilometers`
- Kilometers to Meters:
  `1 kilometer = 1000 meters`
- Miles to Feet:
  `1 mile = 5280 feet`
- Feet to Miles:
  `1 foot = 1 / 5280 miles`

---

2. Weight Conversions

- Grams to Kilograms:
  `1 gram = 0.001 kilograms`
- Kilograms to Grams:
  `1 kilogram = 1000 grams`
- Pounds to Ounces:
  `1 pound = 16 ounces`
- Ounces to Pounds:
  `1 ounce = 1 / 16 pounds`

---

3. Temperature Conversions

- Celsius to Fahrenheit:
  `F = (C × 9/5) + 32`
- Fahrenheit to Celsius:
  `C = (F − 32) × 5/9`
- Celsius to Kelvin:
  `K = C + 273.15`
- Kelvin to Celsius:
  `C = K − 273.15`

Commands to Run the Project

```
cd dotnetapp
dotnet run
dotnet build
```

```
dotnet clean
```

Note

The project will not be submitted if **Submit Project** is not done at least once.

Answer:

```
class UnitConverter
{



   public string GetDefaultUnit(string type)
   {
      switch (type.ToLower())
      {
         case "length": return "meters";
         case "weight": return "kilograms";
         case "temperature": return "celsius";
         default: throw new Exception("Invalid conversion type");
      }
   }
   public double Convert(double value,string fromUnit)
   {
      string type="";
      if (fromUnit == "meters" || fromUnit == "kilometers" || fromUnit == "miles" || fromUnit == "feet")
type = "length";
```

```
    if (fromUnit == "grams" || fromUnit == "kilograms" || fromUnit == "pounds" || fromUnit ==
"ounces") type = "weight";

    if (fromUnit == "celsius" || fromUnit == "fahrenheit" || fromUnit == "kelvin") type = "temperature";


    string defaultUnit = GetDefaultUnit(type);

    return Convert(value,fromUnit,defaultUnit);

  }

  public double Convert(double value,string fromUnit,string toUnit)

  {

    fromUnit = fromUnit.ToLower();

    toUnit = toUnit.ToLower();

    if (fromUnit == toUnit) return value;


    // Length

    if (fromUnit == "meters" && toUnit == "kilometers") return value * 0.001;

    if (fromUnit == "kilometers" && toUnit == "meters") return value * 1000;

    if (fromUnit == "miles" && toUnit == "feet") return value * 5280;

    if (fromUnit == "feet" && toUnit == "miles") return value / 5280;


    // Weight

    if (fromUnit == "grams" && toUnit == "kilograms") return value * 0.001;

    if (fromUnit == "kilograms" && toUnit == "grams") return value * 1000;

    if (fromUnit == "pounds" && toUnit == "ounces") return value * 16;

    if (fromUnit == "ounces" && toUnit == "pounds") return value / 16;


    // Temperature
```

```
        if (fromUnit == "celsius" && toUnit == "fahrenheit") return (value * 9 / 5) + 32;
        if (fromUnit == "fahrenheit" && toUnit == "celsius") return (value - 32) * 5 / 9;
        if (fromUnit == "celsius" && toUnit == "kelvin") return value + 273.15;
        if (fromUnit == "kelvin" && toUnit == "celsius") return value - 273.15;
        throw new Exception("Unsupported conversion");

    }
}
class Program
{
    public static void Main()
    {
        Console.Write("Enter value:");
        double value = double.Parse(Console.ReadLine() ?? "0");

        Console.Write("Enter from unit:");
        string fromUnit = Console.ReadLine() ?? "";

        Console.Write("Enter to unit (press Enter to use default):");
        string toUnit = Console.ReadLine() ?? "";

        UnitConverter converter = new UnitConverter();

        double result = toUnit == "" ? converter.Convert(value, fromUnit) : converter.Convert(value,
fromUnit, toUnit);
```

```
        Console.WriteLine("Result: "+result);


    }
}
```