Question 01

1. Create a C# Console application to convert user given Kilo Meter (km) Value to Meter (m) value. Take a separate Class call “ConvertValues” and inside the class create a method call kilometerTOmeter. (No return type No Parameter Method). And display the answer within the method. Then create an object in main Class (program class) and call the method.

using System;

namespace ConsoleApp22

{

internal class ConvertValues

{

static void Main(string[] args)

{

ConvertValues converter = new ConvertValues();

converter.KilometerToMeter();

Console.ReadLine();

}

public void KilometerToMeter()

{

Console.Write("Enter the distance in kilometers: ");

double kilometers = Convert.ToDouble(Console.ReadLine());

double meters = kilometers \* 1000;

Console.WriteLine("Distance in meters: " + meters);

}

}

}

2. Modify the same user defined method to method which accepts a parameter value. That parameter value is the user given Km value. (No return type with parameter method). Display the answer by using the class object.

using System;

namespace ConsoleApp23

{

internal class ConvertValues

{

static void Main(string[] args)

{

Console.Write("Enter the distance in kilometers: ");

double kilometers = Convert.ToDouble(Console.ReadLine());

ConvertValues converter = new ConvertValues();

converter.KilometerToMeter(kilometers);

Console.ReadLine();

}

public void KilometerToMeter(double kilometers)

{

double meters = kilometers \* 1000;

Console.WriteLine("Distance in meters: " + meters);

}

}

}

3. Modify the same user defined method to method which accept a parameter and returns the answer at the end of the method. You should return the calculated Meter value at the end of the method. (With return type with parameter method). Display the answer by using object.

using System;

namespace ConsoleApp24

{

internal class ConvertValues

{

static void Main(string[] args)

{

Console.Write("Enter the distance in kilometers: ");

double kilometers = Convert.ToDouble(Console.ReadLine());

ConvertValues converter = new ConvertValues();

double meters = converter.KilometerToMeter(kilometers);

Console.WriteLine("Distance in meters: " + meters);

Console.ReadLine();

}

public double KilometerToMeter(double kilometers)

{

double meters = kilometers \* 1000;

return meters;

}

}

}

Question 02

1. Create a C# Console application to find the area and the circumference of a circle.

using System;

namespace ConsoleApp25

{

internal class Program

{

static void Main(string[] args)

{

double radius;

Console.Write("Enter the radius of the circle: ");

if (double.TryParse(Console.ReadLine(), out radius))

{

double area = Math.PI \* Math.Pow(radius, 2);

double circumference = 2 \* Math.PI \* radius;

Console.WriteLine($"Area: {area:F2}");

Console.WriteLine($"Circumference: {circumference:F2}");

}

else

{

Console.WriteLine("Invalid input. Please enter a valid number for the radius.");

}

Console.ReadKey();

}

}

}

2. User should insert the radius value to the program. Program should contain a separate class call “FindValues” inside the separate class add two methods call findArea and findCircumference Both these methods are methods which takes parameters. As the parameter you should pass the radius value. By using above two methods find the area and circumference of the circle and return the answer from both methods. Create a class object in main class and call both methods and display the answers.

using System;

namespace ConsoleApp26

{

internal class FindValues

{

static void Main(string[] args)

{

Console.Write("Enter the radius of the circle: ");

double radius = Convert.ToDouble(Console.ReadLine());

FindValues calculator = new FindValues();

double area = calculator.FindArea(radius);

double circumference = calculator.FindCircumference(radius);

Console.WriteLine("Area of the circle: " + area);

Console.WriteLine("Circumference of the circle: " + circumference);

Console.ReadKey();

}

public double FindArea(double radius)

{

return Math.PI \* radius \* radius;

}

public double FindCircumference(double radius)

{

return 2 \* Math.PI \* radius;

}

}

}