

1

```
namespace GettingInputs
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Enter an integer value");
            int intValue = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine($"{intValue}");

            Console.WriteLine("Enter a float value");
            float floatValue = Convert.ToSingle(Console.ReadLine());
            Console.WriteLine($"{floatValue}");

            Console.WriteLine("Enter a double value");
            double doubleValue = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine($"{doubleValue}");

            Console.WriteLine("Enter a long value");
            long longValue = Convert.ToInt64(Console.ReadLine());
            Console.WriteLine($"{longValue}");

            Console.WriteLine("Enter a string");
            string stringValue = Console.ReadLine();
            Console.WriteLine($"{stringValue}");
        }
    }
}
```

using System;

```
namespace SkyDivingRestrictions
{
    public class Program
```

3

```
    {
        public static void Main(string[] args)
```

```
    {
        Console.WriteLine("Enter your age.");
        int age = Convert.ToInt32(Console.ReadLine());
```

```
        Console.WriteLine("Enter your weight in kilograms.");
        int weight = Convert.ToInt32(Console.ReadLine());
```

```
        if (age >= 18 && weight < 90)
            Console.WriteLine("You are allowed to go skydiving");
        else
```

must

```
            Console.WriteLine("You are not allowed to go skydiving");
        }
    }
}
```

```

namespace Variables //DO NOT change the name
{
    public class Program //DO NOT change the class name
    {
        public static void Main(string[] args) //DO NOT change
        {
            // Get user input
            Console.WriteLine("Enter movie name:");
            string movieName = Console.ReadLine();

            Console.WriteLine("Enter ticket ID:");
            int ticketId = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter seat number:");
            string seatNumber = Console.ReadLine();

            // Create a Movie object
            Movie movie = new Movie();

            // Call method to set movie details
            movie = movie.ReturnMovieDetails(movieName, ticketId, seatNumber);

            // Display the movie details
            Console.WriteLine($"Movie Name: {movie.name}");
            Console.WriteLine($"Ticket ID: {movie.ticketId}");
            Console.WriteLine($"Seat Number: {movie.seatNumber}");
        }
    }
}

```

outside class  
int y = class.x;

→ int x = 5;

```

namespace Variables //DO NOT change the name
{
    public class Movie //DO NOT change the class name
    {
        // Static Variables
        public static string name;

        // Instance Variables
        public int ticketId;
        public string seatNumber;

        // Method to set movie details and return the movie object
        public Movie ReturnMovieDetails(string movieName, int ticketId, string seatNumber)
        {
            Movie name = movieName;
            this.ticketId = ticketId;
            this.seatNumber = seatNumber;
            return this;
        }
    }
}

```

Instance class

↳ class obj = new class();

int y = obj.x;  
or  
this.x denotes instance variable

7

→ this.y  
otherwise  
directly x = y

4

```
using System;
namespace LoanEligibility
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Enter your income:");
            int income = Convert.ToInt32(Console.ReadLine());
```

```
            Console.WriteLine("Enter your credit score:");
            int creditScore = Convert.ToInt32(Console.ReadLine());
```

```
            if (income >= 50000 && creditScore >= 700)
                Console.WriteLine("You are eligible for a loan");
```

```
            else if (income >= 25000 && creditScore >= 650)
                Console.WriteLine("Eligible, but may face higher interest rates");
```

```
            else
                Console.WriteLine("You are not eligible for a loan");
```

```
        }
    }
}
```

5

```
using System;
namespace CaseConverter
{
```

```
    class Program
    {
```

```
        public static void Main(string[] args)
```

```
        {
            Console.WriteLine("Enter the word:");
            string input = Console.ReadLine();
            string result = ConvertCase(input);
            Console.WriteLine(result);
        }
```

```
        static string ConvertCase(string word)
```

```
        {
            char[] characters = word.ToCharArray();
            for (int i = 0; i < characters.Length; i++)
```

```
            {
                if (i % 2 == 0)
```

```
                characters[i] = char.ToUpper(characters[i]);
```

```
            }
            else
                characters[i] = char.ToLower(characters[i]);
```

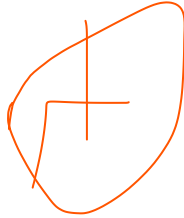
```
            }
            return new string(characters);
        }
```

I → pIl/oa9

O → pIl/Ow

I → Engine

O → EnGiNe



```
using System;
using System.Globalization;

namespace DateEx2
{
    public class Program
    {
        public static string CustomerFeedback(string expectedDate, string installedDate)
        {
            { DateTime expected = DateTime.ParseExact(expectedDate, "MM/dd/yyyy", CultureInfo.InvariantCulture);
              DateTime installed = DateTime.ParseExact(installedDate, "MM/dd/yyyy", CultureInfo.InvariantCulture);
            }

            TimeSpan difference = installed - expected;

            { if (difference.Days < 0)
              return "VeryGood";
            }
            { else if (difference.Days == 0)
              return "Good";
            }
            { else if (difference.Days <= 3)
              return "Average";
            }
            { else
              return "Poor";
            }
        }
    }
}
```

```
public static string FindTheInstalledDay(string installedDate)
{
    { DateTime installed = DateTime.ParseExact(installedDate, "MM/dd/yyyy", CultureInfo.InvariantCulture);
      return installed.DayOfWeek.ToString();
    }
}
```

```
public static void Main(string[] args)
{
    {
    }
}
```

```
Console.WriteLine("Enter the expected date (MM/dd/yyyy):");
string expectedDateInput = Console.ReadLine();
Console.WriteLine("Enter the installed date (MM/dd/yyyy):");
string installedDateInput = Console.ReadLine();
```

```
string installedDay = FindTheInstalledDay(installedDateInput);
string feedback = CustomerFeedback(expectedDateInput, installedDateInput);
Console.WriteLine($"Installed day: {installedDay}");
Console.WriteLine($"Customer Feedback: {feedback}");
```

System;

• text;

• lingq; Enter the character to search;

• Ad. lth

• threading

Tasks;

• Globalization

• Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

6

```
using System;
using System.Text;

namespace InsertingWord //DO NOT change the namespace name
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Enter the word 1");
            String word1 = new StringBuilder(Console.ReadLine());

            Console.WriteLine("Enter the character to search");
            char ch = Console.ReadLine()[0];

            Console.WriteLine("Enter the word 2");
            string word2 = Console.ReadLine();

            Program p = new Program();
            String word1 = p.WordInserting(word1, ch, word2);
            Console.WriteLine(r);
        }
    }
}
```

System;

• text;

• lingq; Enter the character to search;

• Ad. lth

• threading

Tasks;

• Globalization

• Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

• System. Wpf. Text. Regular Expressions;

9

```
string tempWord1 = word1.ToString();
int index = tempWord1.IndexOf(ch);

word1.Insert(index, word2);
return word1;
```



using System;  
using System.Text.RegularExpressions;

namespace RegularExpression

{  
 public class Program

{  
 {

public static void Main(string[] args)

{

Console.WriteLine("Enter the sentence");

string sentence = Console.ReadLine();

~~Console.WriteLine("Enter the starting character");~~

char startChar = Console.ReadLine()[0];

~~Console.WriteLine("Enter the ending character");~~

char endChar = Console.ReadLine()[0];

Program p = new Program();

string[] result = p.FindWords(sentence, startChar, endChar);

Console.WriteLine("The words are below :");

foreach(string word in result)

{  
 Console.WriteLine(word);

}  
 }  
}

public string[] FindWords(string sentence, char startChar, char endChar)

{

string pattern = \$"@\\b{startChar}\\w\*{endChar}\\b";

MatchCollection matches = Regex.Matches(sentence, pattern, RegexOptions.IgnoreCase);

string[] r = new string [matches.Count];

for (int i=0 ; i<matches.Count ; i++)

{

r[i] = matches[i].Value;

}  
 return r;

}  
}

}

8.

⇒

12