**Example 1:** Below example demonstrate the use of regex in Mobile Number Verification. Suppose you are making a form where you need to verify the user-entered mobile number then you can use regex.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_Example1

{

class Program

{

static void Main(string[] args)

{

// Input strings to Match

// valid mobile number

string[] str = { "9925612824",

"8238783138",

"02812451830" };

foreach (string s in str)

{

Console.WriteLine("{0} {1} a valid mobile number.", s,

isValidMobileNumber(s) ? "is" : "is not");

}

Console.ReadKey();

}

// Method containing the regex (fixed indentation)

public static bool isValidMobileNumber(string inputMobileNumber)

{

string strRegex = @"(^[0-9]{10}$)|(^\+[0-9]{2}\s+[0-9]{2}[0-9]{8}$)|(^[0-9]{3}-[0-9]{4}-[0-9]{4}$)";

Regex re = new Regex(strRegex);

if (re.IsMatch(inputMobileNumber))

{

return true;

}

else

{

return false;

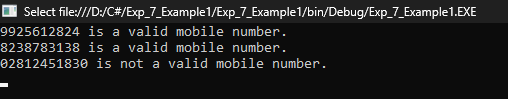
}

}

}

}

**Output:**



**Example 2:** Below example demonstrate the use of regex in Email ID Verification. Suppose you are making a form where you need to verify the user-entered email id then you can use regex.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_2\_Example2

{

class Program

{

static void Main(string[] args)

{

// Input strings for Match

// valid E-mail address.

string[] str = {"parth@gmail.com",

"parthmaniyargmail.com",

"@gmail.com"};

foreach (string s in str)

{

Console.WriteLine("{0} {1} a valid E-mail address.", s,

isValidEmail(s) ? "is" : "is not");

}

Console.ReadKey();

}

// Method to check the Email ID

public static bool isValidEmail(string inputEmail)

{

// This Pattern is use to verify the email

string strRegex = @"\A(?:[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*@(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?)\Z";

Regex re = new Regex(strRegex, RegexOptions.IgnoreCase);

if (re.IsMatch(inputEmail))

return (true);

else

return (false);

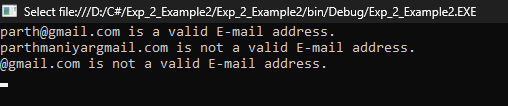
Console.ReadKey();

}

}

}

**Output:**



**Quantifiers:**

**Example 1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_Quantifiers\_1\_2\_3

{

class Program

{

static void Main(string[] args)

{

// pattern b, ab, aab, ...

Regex regex = new Regex(@"a\*b");

Match match = regex.Match("aaaabcd");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

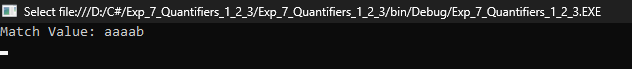
Console.ReadKey();

}

}

}

**Output:**



**Example 2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_Quantifiers\_1\_2\_3

{

class Program

{

static void Main(string[] args)

{

// this will return any pattern

// like ab, aab, aaab, ....

Regex regex = new Regex(@"a+b");

Match match = regex.Match("aaabcd");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

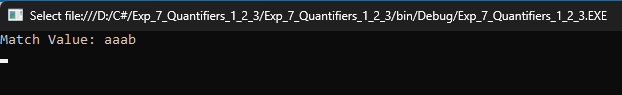
Console.ReadKey();

}

}

}

**Output:**



**Example 3:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_Quantifiers\_1\_2\_3

{

class Program

{

static void Main(string[] args)

{

// This return any pattern like b, ab

Regex regex = new Regex(@"a?b");

Match match = regex.Match("aaaabcd");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

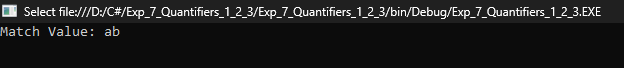
Console.ReadKey();

}

}

}

**Output:**



**Special Characters:**

**Example 1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_SpecialCharacters\_1234

{

class Program

{

static void Main(string[] args)

{

// This will return if shyam exist

// at the beginning of the line

Regex regex = new Regex(@"^Shyam");

Match match = regex.Match("Shyam is my pet name");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

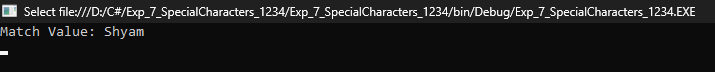
Console.ReadKey();

}

}

}

**Output:**



**Example 2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_SpecialCharacters\_1234

{

class Program

{

static void Main(string[] args)

{

// This return parth if it

// exist at the end of the line

Regex regex = new Regex(@"Parth$");

Match match = regex.Match("My name is Parth");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

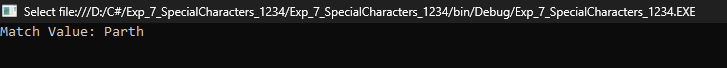
Console.ReadKey();

}

}

}

**Output:**



**Example 3:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_SpecialCharacters\_1234

{

class Program

{

static void Main(string[] args)

{

// This will return any word which

// contains only one letter between

// s and t

Regex regex = new Regex(@"s..t");

Match match = regex.Match("This is my seat");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

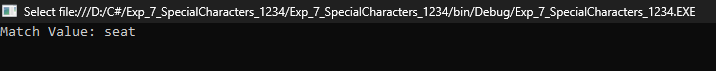
Console.ReadKey();

}

}

}

**Output:**



**Example 4:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_SpecialCharacters\_1234

{

class Program

{

static void Main(string[] args)

{

// This will the return

// the one digit character

Regex regex = new Regex(@"\d");

Match match = regex.Match("I am 19 years old");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

Console.ReadKey();

}

}

}

**Output:**



**Character Classes:**

**Example 1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_CharacterClasses\_123

{

class Program

{

static void Main(string[] args)

{

// This will return one character either

// a or b or c which will come first

Regex regex = new Regex(@"[abc]");

Match match = regex.Match("abcdef");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

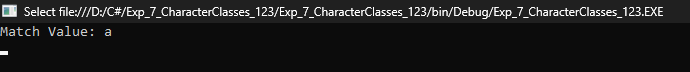
Console.ReadKey();

}

}

}

**Output:**



**Example 2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_CharacterClasses\_123

{

class Program

{

static void Main(string[] args)

{

// This will return any character

// between x and z inclusive

Regex regex = new Regex(@"[x-z]");

Match match = regex.Match("xmax");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

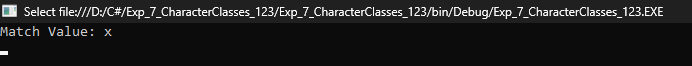
Console.ReadKey();

}

}

}

**Output:**



**Example 3:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_CharacterClasses\_123

{

class Program

{

static void Main(string[] args)

{

// This will return other x,

// y and z character

Regex regex = new Regex(@"[^x-z]");

Match match = regex.Match("xmax");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

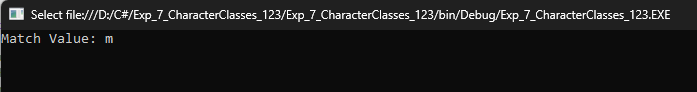
Console.ReadKey();

}

}

}

**Output:**



**Grouping and Alternatives**

**Example 1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_GroupingandAlternatives\_12

{

class Program

{

static void Main(string[] args)

{

// This will return pattern

// will cd, cdcd, cdcdcd, ...

Regex regex = new Regex(@"(cd)+");

Match match = regex.Match("cdcdde");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

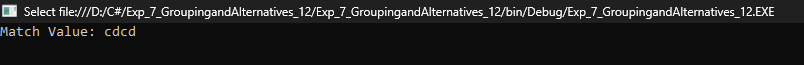
Console.ReadKey();

}

}

}

**Output:**



**Example 1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7\_GroupingandAlternatives\_12

{

class Program

{

static void Main(string[] args)

{

// This will either d or e

// which ever comes first

Regex regex = new Regex(@"d|e");

Match match = regex.Match("edge");

if (match.Success)

{

Console.WriteLine("Match Value: " + match.Value);

}

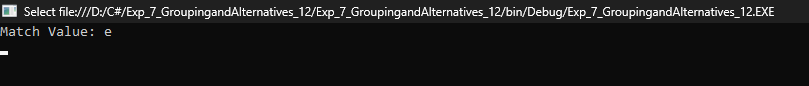
Console.ReadKey();

}

}

}

**Output:**



**Name: Kate Shweta Sanjay**

**Roll No.: 3083**

**Div: B Batch: T4**

**Problem Statement:**

Write Program to validate following data:

EmailID, Mobile No. and Name.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace Exp\_7

{

class Program

{

static void Main(string[] args)

{

// Regular expressions for mobile number, email address, and name

string mobileRegex = @"^\d{10}$";

string emailRegex = @"^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$";

string nameRegex = @"^[a-zA-Z\s]+$";

bool exitRequested = false;

while (!exitRequested)

{

Console.WriteLine("Choose what you want to validate:");

Console.WriteLine("1. Name");

Console.WriteLine("2. Email");

Console.WriteLine("3. Mobile Number");

Console.WriteLine("4. Exit");

int choice;

if (!int.TryParse(Console.ReadLine(), out choice) || choice < 1 || choice > 4)

{

Console.WriteLine("Invalid choice. Please choose a number between 1 and 4.");

continue;

}

string userInput;

switch (choice)

{

case 1:

Console.WriteLine("Enter name:");

userInput = Console.ReadLine();

Console.WriteLine("Name - Valid: " + Regex.IsMatch(userInput, nameRegex));

break;

case 2:

Console.WriteLine("Enter email address:");

userInput = Console.ReadLine();

Console.WriteLine("Email - Valid: " + Regex.IsMatch(userInput, emailRegex));

break;

case 3:

Console.WriteLine("Enter mobile number:");

userInput = Console.ReadLine();

Console.WriteLine("Mobile Number - Valid: " + Regex.IsMatch(userInput, mobileRegex));

break;

case 4:

exitRequested = true;

break;

}

}

Console.WriteLine("Exiting program...");

}

}

}

**Output:**

