C# - Predicate Delegate

A predicate is also a delegate like [Func](http://www.tutorialsteacher.com/csharp/csharp-func-delegate) and [Action](http://www.tutorialsteacher.com/csharp/csharp-action-delegate) delegates. It represents a method that contains a set of criteria and checks whether the passed parameter meets those criteria or not. A predicate delegate methods must take one input parameter and return a boolean - true or false.

The Predicate delegate is defined in the System namespace as shown below:

Predicate signature: public delegate bool Predicate<in T>(T obj);

Same as other delegate types, Predicate can also be used with any method, anonymous method or lambda expression.

Example: Predicate delegate

static bool IsUpperCase(string str)

{

return str.Equals(str.ToUpper());

}

static void Main(string[] args)

{

Predicate<string> isUpper = IsUpperCase;

bool result = isUpper("hello world!!");

Console.WriteLine(result);

}

Output:

false

An anonymous method can also be assigned to a Predicate delegate type as shown below.

Example: Predicate delegate with anonymous method

static void Main(string[] args)

{

Predicate<string> isUpper = delegate(string s)

{

return s.Equals(s.ToUpper());

};

bool result = isUpper("hello world!!");

}

A lambda expression can also be assigned to a Predicate delegate type as shown below.

Example: Predicate delegate with lambda expression

static void Main(string[] args)

{

Predicate<string> isUpper = s => s.Equals(s.ToUpper());

bool result = isUpper("hello world!!");

}

Points to Remember :

1. Predicate delegate takes one input parameter and boolean return type.
2. Predicate delegate must contains some criateria to check whether supplied parameter meets those criateria or not.
3. [Anonymous method](http://www.tutorialsteacher.com/csharp/csharp-anonymous-method) and [Lambda expression](http://www.tutorialsteacher.com/linq/linq-lambda-expression) can be assigned to the predicate delegate.