**C# Exception Filters**

C# Exception Filters is a feature of C# programming language. It is introduced in version C# 6.0. It allows us to specify condition along with a catch block.

C# provides *when* keyword to apply a condition (or filter) along with catch block.

A catch block will execute only when the condition is **true**. If the condition is **false**, catch block is skipped and compiler search for next catch handler.

C# Exception Filters is used for logging purpose.

C# Exception Filter Syntax

1. **catch** (ArgumentException e) when (e.ParamName == "?"){  }

In the following example, we are implementing exception filter. It executes only, if compiler throws an **IndexOutOfRangeException** exception.

C# Exception Filter Example

**using** System;

**namespace** CSharpFeatures

{

**class** ExceptionFilter

    {

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

        {

**try**

            {

**int**[] a = **new** **int**[5];

                a[10] = 12;

            }**catch**(Exception e) when(e.GetType().ToString() == "System.IndexOutOfRangeException")

            {

                // Executing some other task

                SomeOtherTask();

            }

        }

**static** **void** SomeOtherTask()

        {

            Console.WriteLine("A new task is executing...");

        }

    }

}

Output

*A new task is executing...*

In the following example, we are throwing exception explicitly that matches with *when* condition.

C# Exception Filter Example2

**using** System;

**namespace** CSharpFeatures

{

**class** ExceptionFilter

    {

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

        {

**try**

            {

                // Throwing Exception explicitly

**throw** **new** IndexOutOfRangeException("Array Exception Occured");

            }**catch**(Exception e) when(e.Message == "Array Exception Occured")

            {

                Console.WriteLine(e.Message);

                SomeOtherTask();

            }

        }

**static** **void** SomeOtherTask()

        {

            Console.WriteLine("A new task is executing...");

        }

    }

}

Output

*Array Exception Occured*

*A new task is executing...*