



Programming Language - C#

Exception handling





Exceptions

- An exception is an erroneous situation that occurs during program execution.
- When an exception occurs in an application, the system throws an error.
- The error is handled through the process of exception handling.



Exception handling

- Unlike java , C# supports only unchecked exceptions
- The .NET framework provides built in classes for common exceptions
- Exceptions are enclosed in try catch and finally block
- In C# the catch and finally block are optional.
- In C# exceptions are nothing but objects of type Exception
- C# does not support throws keyword



Unchecked Exceptions

- The exceptions that are not checked during compile time
- Errors that occur during the execution of the program , will stop the program execution abruptly, hence it has to be handled.
- The exceptions are fired due to some abnormal developer's logical error.



Exception Classes

- There are many exception classes which are directly or indirectly derived from the `System.Exception` class. Some of these classes are:

Exception Classes	Description
<code>System.IO.IOException</code>	Handles I/O errors
<code>System.IndexOutOfRangeException</code>	Handles errors generated when a method refers to an array element, which is out of its bound
<code>System.NullReferenceException</code>	Handles errors generated during the process of dereferencing a null object
<code>System.DivideByZeroException</code>	Handles errors generated during the process of dividing the dividend with zero
<code>System.InvalidCastException</code>	Handles errors generated during typecasting
<code>System.OutOfMemoryException</code>	Handles memory allocation to the application errors

Contd..

```
try
{
    //statements that may cause an exception
}
catch (...)
{
    //error handling code
}
finally
{
    //statements to be executed
}
```



Assignment – Exception Handling

David is working on a project where he is calculating the sum of values in an integer array. David needs to handle the exceptions, which can occur while he is working with the arrays. If any exceptional condition is reached when David is executing the application, the application needs to display an exception message.

Help David to handle the exceptions.

User-Defined Exceptions



- In C#, you can create your own exception class. Such kinds of exceptions are known as user-defined exceptions or application exception.
- The Exception class is the base class for all the exceptions in C#.
- The user defined exceptions classes must inherit from either Exception class or one of its standard derived classes

Steps to create a user defined exception



Inherit a class from the exception class

```
Public class CustomerNotFoundException:Exception
```

Override the string message from the exception class

```
public override string Message  
{  
    get{ Return "customer not found" }  
}
```

Raise the exception

```
Throw new CustomerNotFoundException();
```

Assignment – User defined exception



- You are working for ICICI bank as a developer.
- You're working on the fund transfer module.
- You need to handle the situation when the customer wishes to transfer more money than he has in his/her account.



Thank you

Innovative Services



Passionate Employees



Delighted Customers

