**EXCEPTION HANDLING**

**EXCEPTION:**

It is runtime logical error that interrupts program execution. It may or may not happen.

**EXCEPTION HANDLING:**

Exception handling is an in built mechanism in .NET framework to detect and handle run time errors. The .NET framework contains lots of standard exceptions. The exceptions are anomalies that occur during the execution of a program. They can be because of user, logic or system errors.

NameSpace: System.Exception

**TYPES:**

* Pre-defined
* User-defined
  1. Checked.
  2. Unchecked.

**EXCEPTION HANDLING:**

* Try
* Catch
* Throw
* Finally

**Try:** A try block identifies a block of code for which particular exceptions will be activated. It's followed by one or more catch blocks.

**Catch**: A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.

**Finally**: The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example, if you open a file, it must be closed whether an exception is raised or not.

**Throw**: A program throws an exception when a problem shows up. This is done using a throw keyword.

**SYNTAX:**

try  
{  
// Statement which can cause an exception.  
}  
catch(Type x)  
{  
// Statements for handling the exception

// throw X  
}  
finally  
{  
//Any cleanup code  
}