

Answer on Question#38073- Programming, C#

1. Sharon has started to use both the stock details management and the file information viewer features of the application. However, while using the file information viewer application, she accidentally enters the path of a folder that does not exist on her computer. This caused the application to stop responding. In addition, the format in which the stock details are displayed is not clear. Therefore, she asks Hayley to modify the application so that: The file information viewer does not stop responding in case a specified folder does not exist. The stock details management application adds a line break before every stock details entry whenever a new entry is added to the stock details file.

Solution.

The try-catch statement consists of a **try** block followed by one or more **catch** clauses, which specify handlers for different exceptions.

Remarks

When an exception is thrown, the common language runtime (CLR) looks for the **catch** statement that handles this exception. If the currently executing method does not contain such a **catch** block, the CLR looks at the method that called the current method, and so on up the call stack. If no **catch** block is found, then the CLR displays an unhandled exception message to the user and stops execution of the program. The **try** block contains the guarded code that may cause the exception. The block is executed until an exception is thrown or it is completed successfully. For example, the following attempt to cast a **null** object raises the `NullReferenceException` exception:

```
C#
object o2 = null;
try
{
    int i2 = (int)o2;    // Error
}
```

Although the **catch** clause can be used without arguments to catch any type of exception, it is not recommended. In general, you should only catch those exceptions you know how to recover from. Therefore, you should always specify an object argument derived from `System.Exception` For example:

```
C#
catch (InvalidCastException e)
{
}
```

It is possible to use more than one specific **catch** clause in the same try-catch statement. In this case, the order of the **catch** clauses is important because the **catch** clauses are examined in order. Catch is more specific exception before the less specific one. The compiler produces an error if you order your catch blocks so that a later block can never be reached.