private void dataGridView1\_CellValidating(object sender,

DataGridViewCellValidatingEventArgs e)

{

dataGridView1.Rows[e.RowIndex].ErrorText = "";

int newInteger;

// Don't try to validate the 'new row' until finished

// editing since there

// is not any point in validating its initial value.

if (dataGridView1.Rows[e.RowIndex].IsNewRow) { return; }

if (!int.TryParse(e.FormattedValue.ToString(),

out newInteger) || newInteger < 0)

{

e.Cancel = true;

dataGridView1.Rows[e.RowIndex].ErrorText = "the value

**private void dataGridView1\_CellValidating(object sender,**

**DataGridViewCellValidatingEventArgs e)**

**{**

**string headerText =**

**dataGridView1.Columns[e.ColumnIndex].HeaderText;**

**// Abort validation if cell is not in the CompanyName column.**

**if (!headerText.Equals("CompanyName")) return;**

**// Confirm that the cell is not empty.**

**if (string.IsNullOrEmpty(e.FormattedValue.ToString()))**

**{**

**dataGridView1.Rows[e.RowIndex].ErrorText =**

**"Company Name must not be empty";**

**e.Cancel = true;**

**}**

**}**

**void dataGridView1\_CellEndEdit(object sender, DataGridViewCellEventArgs e)**

**{**

**// Clear the row error in case the user presses ESC.**

**dataGridView1.Rows[e.RowIndex].ErrorText = String.Empty;**

**Private Sub DataGridView1\_CellEndEdit(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellEndEdit**

**If (e.ColumnIndex = 0) Then ' Checking numeric value for Column1 only**

**Dim value As String = DataGridView1.Rows(e.RowIndex).Cells(e.ColumnIndex).Value.ToString()**

**If Not Information.IsNumeric(value) Then**

**MessageBox.Show("Please enter numeric value.")**

**DataGridView1.Rows(e.RowIndex).Cells(e.ColumnIndex).Value = String.Empty**

**Exit Sub**

**End If**

**End If**

**End Sub**