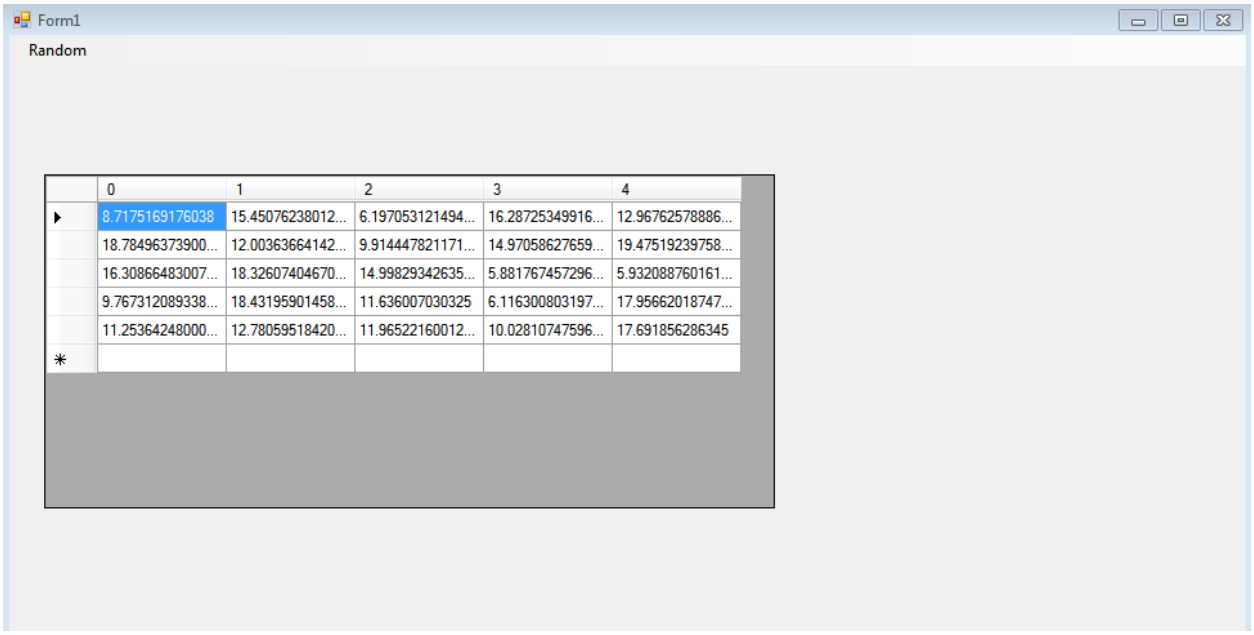
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace MyRandomNumbers
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void integersToolStripMenuItem_Click(object sender, EventArgs e)
        {
            int[,] x = new int[5, 5];
            Random rnd = new Random();
            for (int i = 0; i < x.GetLength(0); i++)
            {
                for (int j = 0; j < x.GetLength(1); j++)
                {
                    //x[i, j] = rnd.Next();  // any non-negative integer
                    //x[i, j] = rnd.Next(100);  //(0-99) non negative integer
                    x[i, j] = rnd.Next(40,100);   //(40-99) non negative integer
                }
            }
            DataTable dt = new DataTable();
            for (int i = 0; i < x.GetLength(1); i++)
            {
                dt.Columns.Add();
            }
            DataRow dr;
            for (int i = 0; i < x.GetLength(0); i++)
            {
                dr = dt.NewRow();
                for (int j = 0; j < x.GetLength(1); j++)
                {
                    dr[j] = x[i, j];
                }
                dt.Rows.Add(dr);
            }
            dataGridView1.DataSource = dt;
        }

        private void doubleToolStripMenuItem_Click(object sender, EventArgs e)
        {

           double[,] x = new double[5, 5];
            Random rnd = new Random(2);          //passing a seed value
            for (int i = 0; i < x.GetLength(0); i++)
            {
                for (int j = 0; j < x.GetLength(1); j++)
```

```
                {
                    x[i, j] = rnd.NextDouble();
                }
            }

            DataTable dt = new DataTable();
            for (int i = 0; i < x.GetLength(1); i++)
            {
                dt.Columns.Add(i.ToString());
            }
            DataRow dr;
            for (int i = 0; i < x.GetLength(0); i++)
            {
                dr = dt.NewRow();
                for (int j = 0; j < x.GetLength(1); j++)
                {
                    dr[j] = x[i, j];
                }
                dt.Rows.Add(dr);
            }
            dataGridView1.DataSource = dt;
        }
        //Display double values between 5 to 20
        private void doubleRangeToolStripMenuItem_Click(object sender, EventArgs e)
        {

            double[,] x = new double[5, 5];
            Random rnd = new Random();
            for (int i = 0; i < x.GetLength(0); i++)
            {
                for (int j = 0; j < x.GetLength(1); j++)
                {
                    x[i, j] = rnd.NextDouble() + rnd.Next(5, 20);
                }
            }
            DataTable dt = new DataTable();
            for (int i = 0; i < x.GetLength(1); i++)
            {
                dt.Columns.Add(i.ToString());
            }
            DataRow dr;
            for (int i = 0; i < x.GetLength(0); i++)
            {
                dr = dt.NewRow();
                for (int j = 0; j < x.GetLength(1); j++)
                {
                    dr[j] = x[i, j];
                }
                dt.Rows.Add(dr);
            }
            dataGridView1.DataSource = dt;
        }

    }
}
```


Form1
Random
0
1
2
3
4
8.7175169176038
15.45076238012...
6.197053121494...
16.28725349916...
12.96762578886...
18.78496373900...
12.00363664142...
9.914447821171...
14.97058627659...
19.47519239758...
16.30866483007...
18.32607404670...
14.99829342635...
5.881767457296...
5.932088760161...
9.767312089338...
18.43195901458...
11.636007030325
6.116300803197...
17.95662018747...
11.25364248000...
12.78059518420...
11.96522160012...
10.02810747596...
17.691856286345