**Name:** **Session:**

# Programming II
# Lab Exercise 3.20.2024
# Plotting Math Functions in C#

In this lab, you will be learning to create graphs of mathematical functions using the Chart control. In the first exercise, you will be plotting the sin and cosine functions. In Visual C# as in most programming language, the trigonometric functions will require you to provide those functions (sin and cos) arguments that are in radians (not degrees). You will also learn how to send your window directly to the printer using the PrintForm control which can be found in the Visual Basic PowerPacks section of the Toolbox.

**Project 1 – Sine and Cosine plotter**

1. Create a new Windows Form application

2. Add a Chart control (found in the Data section of the Toolbox). You will need to configure the Chart control properties by adding Series 2 to the series collection. Also set Series to so that its color is red. Make sure both Series 1 and Series 2 are line charts.

3. Add 3 Button controls (Plot, Print, and Exit). Name your buttons btnPlot, btnPrint, and btnExit.

4. Add a PrintForm control (found in the Visual Basic PowerPacks section of the Toolbox)

5. Add the following code to the btnPrint_Click procedure:
```
printForm1.Print();
```

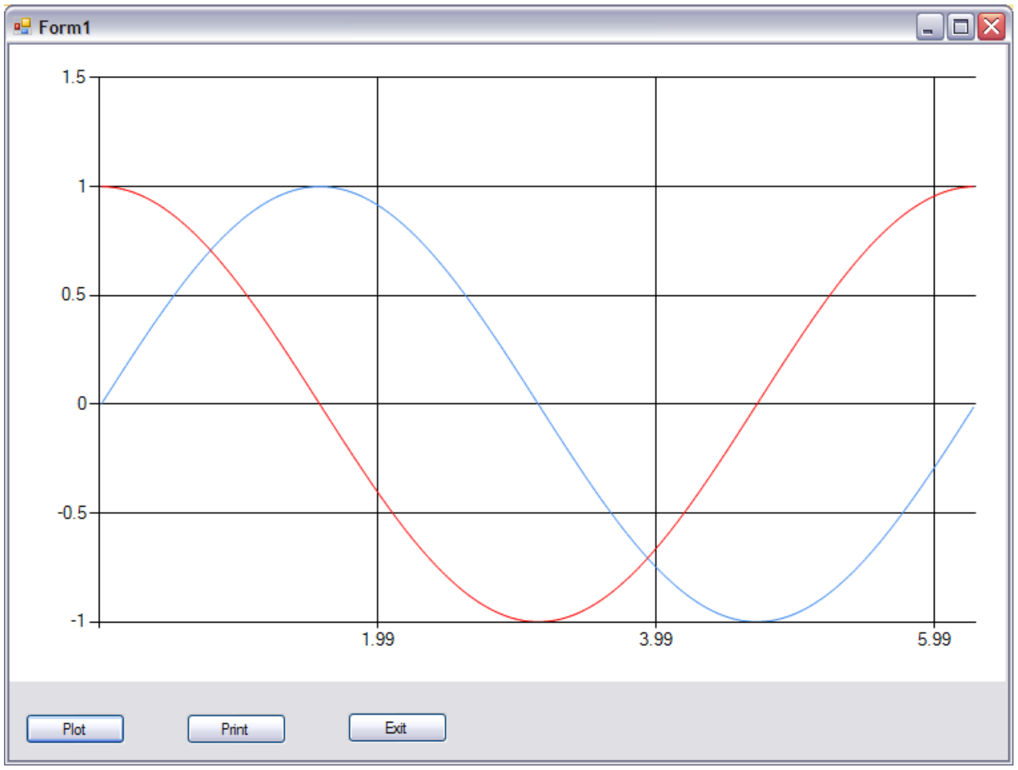
6. Add the following code to the btnExit_Click procedure:
```
this.Close();
```

7. Add the following code to the btnPlot_Click procedure:

```
double[] x = new double[360];
double[] y = new double[360];
double[] y2 = new double[360];

for (int i = 0; i <= 359; i++)
{
        x[i] = Convert.ToDouble((i * Math.PI / 180).ToString("f2"));
        y[i] = Math.Sin(x[i]);
        chart1.Series[0].Points.AddXY(x[i], y[i]);
        y2[i] = Math.Cos(x[i]);
        chart1.Series[1].Points.AddXY(x[i], y2[i]);
}
```

8. Run the program and print the form using the Print button. You should get something like this:



**Project 2 – Order 4 Polynomial Plotter**

1. Create a new Windows Form Application
2. Design your interface to allow the user to enter up to a 4th order polynomial
3. Your application should have a Plot, Print, and Exit
4. Create a plot of the function in the range of -10 to 10:

$$f(x) = 3x^4 + 7x^3 - 2x^2 + 13x - 6$$

5. Add the following code to the btnPlot_Click event handler.

```
//Declare variables
double[] x = new double[1001];
double[] y = new double[1001];
int ex;
double x0, x1, x2, x3, x4;
```

```
//If any TextBox is empty (contains "") assign "0" to TextBox
if (txtX0.Text == "")
   txtX0.Text = "0";
if (txtX1.Text == "")
   txtX1.Text = "0";
if (txtX2.Text == "")
   txtX2.Text = "0";
if (txtX3.Text == "")
   txtX3.Text = "0";
if (txtX4.Text == "")
   txtX4.Text = "0";

//Convert all TextBoxes Text to double and store in x0, x1, x2, x3, x4, and x5
x0 = Convert.ToDouble(txtX0.Text);
x1 = Convert.ToDouble(txtX1.Text);
x2 = Convert.ToDouble(txtX2.Text);
x3 = Convert.ToDouble(txtX3.Text);
x4 = Convert.ToDouble(txtX4.Text);

//Generate 1001 points at intervals of 0.02
for (int i = 0; i <= 1000; i++)
{
   ex = (i - 500) / 50;
   x[i] = ex;
   y[i] = x4 * Math.Pow(ex,  4) + x3 * Math.Pow(ex, 3) + x2 * ex * ex + x1 * ex + x0;
   chart1.Series[0].Points.AddXY(x[i], y[i]);
}
```

6. Add the following code to the btnPrint_Click event

```
printForm1.Print();
```

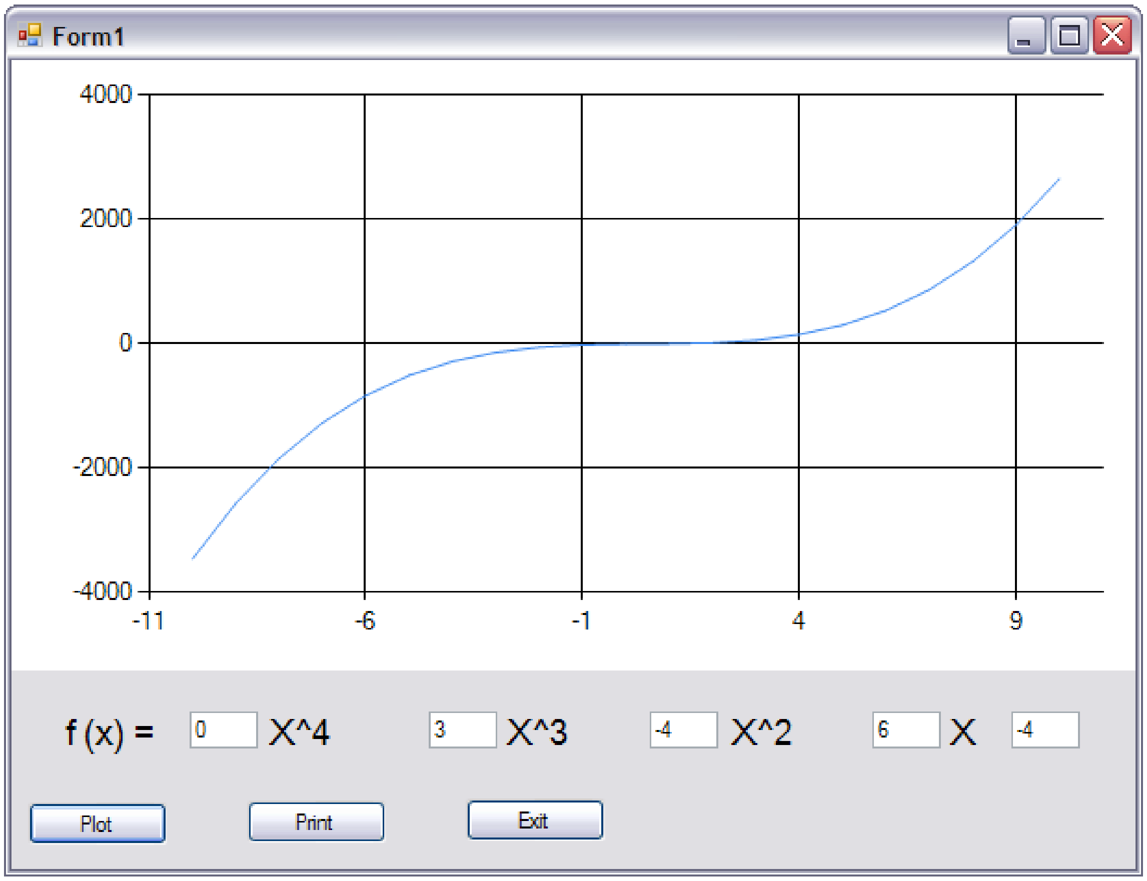
7. Add the following code to the btnExit_Click event

```
this.Close();
```

8. Add the following code to the btnClear_Click event

```
txtX0.Text = "";
txtX1.Text = "";
txtX2.Text = "";
txtX3.Text = "";
txtX4.Text = "";
chart1.Series[0].Points.Clear();
txtX4.Focus();
```

Here is a screenshot of my interface



f (x) = 0 X^4 3 X^3 -4 X^2 6 X -4

When you have completed this project, print a copy of your running program window that is plotting the function $f(x) = 2x^3 + 7x^2 - 3x + 9$.