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

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

        private void simpleHarmonicMotionToolStripMenuItem_Click(object sender, EventArgs e)
        {

        }

        private void idealSimplePendulumToolStripMenuItem_Click(object sender, EventArgs e)
        {

        }

        private void eulerMethodToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Refresh();
            float th, om;
            th = float.Parse(textBox1.Text);
            om = float.Parse(textBox2.Text);
            //create oscillator object
            Oscillator obj = new Oscillator(th, om);
            GraphicalSetup gs = new GraphicalSetup(this);
            gs.DrawAxes(gs.x0, gs.y0, "t", "omega,  theta", 0, 0);
            //Make the Pendulum
            //suspension line
            float leng = 200, left = 20, pleng = 100, up = 20;

            Point p1 = new Point((int)(2 * gs.x0 - leng - left), (int)(2 * gs.y0 - pleng - up));
            Point p2 = new Point((int)(2 * gs.x0 - left), (int)(2 * gs.y0 - pleng - up));
            gs.gg.DrawLine(gs.pblue, p1, p2);
            //suspension point
            float xs = (p1.X + p2.X) / 2, ys = p1.Y;
            //suspend the pendulum
            gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
                ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
            while (obj.t < 40)
            {
                gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
               ys + pleng * (float)Math.Cos(obj.th));
                //fix the bob
```

```
            gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
            System.Threading.Thread.Sleep(1);
            gs.gg.DrawLine(gs.pwhite, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
           ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bwhite, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);


            gs.plotter(obj.t, obj.th, 7, 0.4f, gs.bblue);
            gs.plotter(obj.t, obj.om, 7, 0.4f, gs.bred);
            //gs.plotter(obj.t, obj.TotalEnergy(), 10, 12,gs.bred);
            obj.IdealOscillateEuler();
            //textBox1.Text = obj.th.ToString() + "     " + obj.om.ToString();
            //textBox1.Refresh();

        }
        gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
         ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
            ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);

    }

    private void cromerMethodToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Refresh();
        float th, om;
        th = float.Parse(textBox1.Text);
        om = float.Parse(textBox2.Text);
        //create oscillator object
        Oscillator obj = new Oscillator(th, om);
        GraphicalSetup gs = new GraphicalSetup(this);
        gs.DrawAxes(gs.x0, gs.y0, "t", "omega,  theta", 0, 0);
        //Make the Pendulum
        //suspension line
        float leng=200,left=20,pleng=100,up=20;

        Point p1 = new Point((int)(2 * gs.x0 - leng - left), (int)(2 * gs.y0 - pleng - up));
        Point p2 = new Point((int)(2 * gs.x0 - left), (int)(2 * gs.y0 - pleng - up));
        gs.gg.DrawLine(gs.pblue, p1, p2);
        //suspension point
        float xs = (p1.X + p2.X) / 2, ys = p1.Y;
        //suspend the pendulum
        gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
            ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th)-7,
            ys + pleng * (float)Math.Cos(obj.th)-7, 15, 15);
        while (obj.t < 40)
        {
            gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
           ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
```

```
            System.Threading.Thread.Sleep(1);
            gs.gg.DrawLine(gs.pwhite, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
           ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bwhite, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);


            gs.plotter(obj.t, obj.th, 7, 0.4f, gs.bblue);
            gs.plotter(obj.t, obj.om, 7, 0.4f, gs.bred);
            //gs.plotter(obj.t, obj.TotalEnergy(), 10, 12,gs.bred);
            obj.IdealOscillateCromer();
            //textBox1.Text = obj.th.ToString() + "     " + obj.om.ToString();
            //textBox1.Refresh();

        }
        gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
         ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
            ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);

    }

    private void eulerMethodToolStripMenuItem1_Click(object sender, EventArgs e)
    {

    }

    private void dampedOscillationsToolStripMenuItem_Click(object sender, EventArgs e)
    {
      //  this.Refresh();
        float th, om,q,FD,omD;
        th = float.Parse(textBox1.Text);
        om = float.Parse(textBox2.Text);
        q = float.Parse(textBox3.Text);
        FD = float.Parse(textBox4.Text);
        omD = float.Parse(textBox5.Text);
        //create oscillator object
        Oscillator obj = new Oscillator(th, om,q,FD,omD);
        GraphicalSetup gs = new GraphicalSetup(this);
        gs.DrawAxes(gs.x0, gs.y0, "t", "omega,  theta", 0, 0);
        //Make the Pendulum
        //suspension line
        float leng = 200, left = 20, pleng = 100, up = 20;

        Point p1 = new Point((int)(2 * gs.x0 - leng - left), (int)(2 * gs.y0 - pleng - up));
        Point p2 = new Point((int)(2 * gs.x0 - left), (int)(2 * gs.y0 - pleng - up));
        gs.gg.DrawLine(gs.pblue, p1, p2);
        //suspension point
        float xs = (p1.X + p2.X) / 2, ys = p1.Y;
        //suspend the pendulum
        gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
            ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
            ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
        while (obj.t < 10)
```

```
    {
        gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
       ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
            ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
        System.Threading.Thread.Sleep(1);
        gs.gg.DrawLine(gs.pwhite, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
       ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bwhite, xs + pleng * (float)Math.Sin(obj.th) - 7,
            ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);

        if(obj.q==1)
        gs.plotter(obj.t, obj.th, 50, 150, gs.bblue);
        if (obj.q == 5)
            gs.plotter(obj.t, obj.th, 50, 150, gs.bred);
        if (obj.q == 10)
            gs.plotter(obj.t, obj.th, 50, 150, gs.bgreen);
       // gs.plotter(obj.t, obj.om, 7, 20, gs.bred);
        //gs.plotter(obj.t, obj.TotalEnergy(), 10, 12,gs.bred);
        obj.Damped();
        //textBox1.Text = obj.th.ToString() + "     " + obj.om.ToString();
        //textBox1.Refresh();

    }
    gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
     ys + pleng * (float)Math.Cos(obj.th));
    //fix the bob
    gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
        ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);

}

private void dampedDrivenOscillationsToolStripMenuItem_Click(object sender, EventArgs e)
{
    //  this.Refresh();
    float th, om, q, FD, omD;
    th = float.Parse(textBox1.Text);
    om = float.Parse(textBox2.Text);
    q = float.Parse(textBox3.Text);
    FD = float.Parse(textBox4.Text);
    omD = float.Parse(textBox5.Text);
    //create oscillator object
    Oscillator obj = new Oscillator(th, om, q, FD, omD);
    GraphicalSetup gs = new GraphicalSetup(this);
    gs.DrawAxes(gs.x0, gs.y0, "t", "omega,  theta", 0, 0);
    //Make the Pendulum
    //suspension line
    float leng = 200, left = 20, pleng = 100, up = 20;

    Point p1 = new Point((int)(2 * gs.x0 - leng - left), (int)(2 * gs.y0 - pleng - up));
    Point p2 = new Point((int)(2 * gs.x0 - left), (int)(2 * gs.y0 - pleng - up));
    gs.gg.DrawLine(gs.pblue, p1, p2);
    //suspension point
    float xs = (p1.X + p2.X) / 2, ys = p1.Y;
    //suspend the pendulum
    gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
```

```
            ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
            ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
        while (obj.t < 60)
        {
            gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
           ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
            System.Threading.Thread.Sleep(1);
            gs.gg.DrawLine(gs.pwhite, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
           ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bwhite, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);


                gs.plotter(obj.t, obj.th, 10, 150, gs.bblue);
                gs.plotter(obj.th, obj.om, 100, 100, gs.bred);
                if (obj.th < -Math.PI) obj.th = obj.th + 2 * (float)Math.PI;
                if (obj.th > Math.PI) obj.th = obj.th - 2 * (float)Math.PI;
            // gs.plotter(obj.t, obj.om, 7, 20, gs.bred);
            //gs.plotter(obj.t, obj.TotalEnergy(), 10, 12,gs.bred);
            obj.DampedDriven();
            //textBox1.Text = obj.th.ToString() + "     " + obj.om.ToString();
            //textBox1.Refresh();

        }
        gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
         ys + pleng * (float)Math.Cos(obj.th));
        //fix the bob
        gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
            ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);


    }

    private void nonLinearDampedDrivenOscillationsToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        //  this.Refresh();
        float th, om, q, FD, omD;
        th = float.Parse(textBox1.Text);
        om = float.Parse(textBox2.Text);
        q = float.Parse(textBox3.Text);
        FD = float.Parse(textBox4.Text);
        omD = float.Parse(textBox5.Text);
        //create oscillator object
        Oscillator obj = new Oscillator(th, om, q, FD, omD);
        GraphicalSetup gs = new GraphicalSetup(this);
        gs.DrawAxes(gs.x0, gs.y0, "t", "omega,  theta", 0, 0);
        //Make the Pendulum
        //suspension line
        float leng = 200, left = 20, pleng = 100, up = 20;

        Point p1 = new Point((int)(2 * gs.x0 - leng - left), (int)(2 * gs.y0 - pleng - up));
```

```
            Point p2 = new Point((int)(2 * gs.x0 - left), (int)(2 * gs.y0 - pleng - up));
            gs.gg.DrawLine(gs.pblue, p1, p2);
            //suspension point
            float xs = (p1.X + p2.X) / 2, ys = p1.Y;
            //suspend the pendulum
            gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
                ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
            float omDto2Pi,check;
            while (obj.t < 90)
            {
                gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
               ys + pleng * (float)Math.Cos(obj.th));
                //fix the bob
                gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
                    ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
                //System.Threading.Thread.Sleep(1);
                gs.gg.DrawLine(gs.pwhite, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
               ys + pleng * (float)Math.Cos(obj.th));
                //fix the bob
                gs.gg.FillEllipse(gs.bwhite, xs + pleng * (float)Math.Sin(obj.th) - 7,
                    ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);
                omDto2Pi = obj.omD*obj.t / (2 * (float)Math.PI);
                check = omDto2Pi - (int)omDto2Pi;



                gs.plotter(obj.t, obj.th, 5, 15, gs.bblue);
                //if (Math.Abs(check) < 0.001 || Math.Abs(check) > 0.999)
                gs.plotter(obj.th, obj.om, 100, 100, gs.bred);
                gs.plotter(obj.t, obj.om, 7, 20, gs.bred);
                //gs.plotter(obj.t, obj.TotalEnergy(), 10, 12,gs.bred);
                obj.Chaotic();
                if (obj.th < -Math.PI) obj.th = obj.th +2*(float)Math.PI;
                if (obj.th >Math.PI) obj.th = obj.th - 2*(float)Math.PI;
                //textBox1.Text = obj.th.ToString() + "     " + obj.om.ToString();
                //textBox1.Refresh();

            }
            gs.gg.DrawLine(gs.pblue, xs, ys, xs + pleng * (float)Math.Sin(obj.th),
             ys + pleng * (float)Math.Cos(obj.th));
            //fix the bob
            gs.gg.FillEllipse(gs.bblue, xs + pleng * (float)Math.Sin(obj.th) - 7,
                ys + pleng * (float)Math.Cos(obj.th) - 7, 15, 15);



        }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;
//using System.Threading.Tasks;
using System.Drawing;

namespace SimpleHarmonicMotion
{
    class GraphicalSetup
    {
        //Data
        public Graphics gg;
        public SolidBrush bred, bblue, bwhite,bgreen;
      public  Pen pblue, pred, pwhite;
       public Font f;
        public float x0, y0;
        //constructor
        public GraphicalSetup(Form1 frm)
        {
            gg = frm.CreateGraphics();
            bred = new SolidBrush(Color.Red);
            bblue = new SolidBrush(Color.Blue);
            bgreen = new SolidBrush(Color.Green);
            bwhite = new SolidBrush(Color.White);
            pred = new Pen(Color.Red);
            pblue = new Pen(Color.Blue,3);
            pwhite = new Pen(Color.White,3);
            f = new Font("Arial",16);
            x0 = frm.ClientSize.Width / 2;
            y0 = frm.ClientSize.Height / 2;
        }
        //Other Functions
        public void DrawAxes(float xlen,float ylen,
            string xlabel,string ylabel,float deltax,
            float deltay)
        {
            gg.DrawLine(pred, x0, y0, x0 + xlen, y0);
            gg.DrawLine(pred, x0, y0, x0, y0 - ylen);
            gg.DrawString(xlabel,f,bblue,
                x0 + xlen / 2, y0 + 10);
            gg.DrawString(ylabel, f, bblue,
               x0 -150, y0-ylen/2);

        }
        public void plotter(float hv,float vv,
            float hscale,float vscale,SolidBrush b)
        {
            gg.FillEllipse(b, x0 + hv * hscale,
                y0 - vv * vscale-5, 5, 5);
        }
        public void eraser(float hv, float vv,
            float hscale, float vscale)
        {
            gg.FillEllipse(bwhite, x0 + hv * hscale,
                y0 - vv * vscale-5, 5, 5);
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace SimpleHarmonicMotion
{
    class Oscillator
    {
        //Data variables
        public float th, om, t, dt, L, g, FD, omD, q, T,m;
        public int n;
        //Constructor
        public Oscillator(float theta, float omega)
        {//Constructor for Ideal Simple Pendulum
            th = theta; om = omega; t = 0; dt = 0.04f; L = 1; g = 9.8f; m = 1;
        }
        public Oscillator(float theta, float omega, float q, float FD, float omD)
        {//Constructorr for Realistic Simple Pendulum
            th = theta; om = omega; t = 0; dt = 0.04f; L =9.8f; g = 9.8f; this.FD = FD;
            this.omD = omD; this.q = q; m = 1;
        }
        //other functions
        public void IdealOscillateEuler()
        {
            th = th + om * dt;
            om = om - (g / L) * th * dt;
             t = t + dt;
        }
        public void IdealOscillateCromer()
        {

            om = om - (g / L) * th * dt;
            th = th + om * dt;
            t = t + dt;
        }
        public void Damped()
        {

            om = om - ((g / L) * th+q*om) * dt;
            th = th + om * dt;
            t = t + dt;
        }
        public void DampedDriven()
        {

            om = om - ((g / L) * th+q*om-
                FD*(float)Math.Sin(omD*t)) * dt;
            th = th + om * dt;
            t = t + dt;
        }
        public void Chaotic()
        {

            om = om - ((g / L) *(float)Math.Sin(th)+q*om-
                FD*(float)Math.Sin(omD*t)) * dt;
            th = th + om * dt;
            t = t + dt;
```

```
        }
        public float TotalEnergy()
        {
            return (1/2*m*L*L*om*om+m*g*L*(1-(float)Math.Cos(th)));
        }



    }
}
```