

## foreach

```
foreach (DataRow drow in table.Rows)
{
    ...
}
```

## ArrayList (System.Collections.Hashtable)

```
ArrayList list = new ArrayList();
string s = "hello";
list.Add(s);
```

## Hashtable (System.Collections.Hashtable)

```
Hashtable htable = new Hashtable();
htable.Add("hello", 10);
if (htable.Contains("hello"))
{
    int val = (int) htable["hello"];
}
```

## Programmatically Adding Controls

```
Label lbl = new Label();
lbl.Location = new Point(5, 10);
this.Controls.Add(mylabel);
```

## Pausing / Adding Delay

```
using System.Threading;
public void PauseForALittle()
{
    Thread.Sleep(1000);
}
```

## Converting Ints to String

```
int num = 13;
num.ToString();
```

## Converting Strings to Ints

```
int num = int.Parse("13");
```

## Comparing String Values

```
string s = "hello";
string t = "hello";
if (s == t)
{
    //s and t have the same value
    //s.Equals(t) is the same thing
}
```

## Interacting Between Forms

```
In Form1.cs
SecondForm secondform = new SecondForm(this);

In SecondForm.cs
public SecondForm(System.Windows.Forms.Form p)
{
    parentForm = p;
}

private void SecondForm(object sender, CancelEventArgs e)
{
    parentForm.secondFormClosing();
}
```

## Painting a Gradient

```
private void CustomControl_Paint(object s, PaintEventArgs e)
{
    Color b = Color.LightBlue;
    Color w = Color.White;
    Rectangle r = new Rectangle(0, 0, this.Width, this.Height);
    LinearGradientBrush b = new LinearGradientBrush(r, b, w, 90);
    e.Graphics.FillRectangle(b, rect);
}
```

## Timer

```
Timer minTimer = new Timer(); // or drag from Form Designer

minTimer.Interval = 60000;
minTimer.Tick += new System.EventHandler(this.minTimer_Tick);
minTimer.Start();

private void minTimer_Tick(object sender, System.EventArgs e)
{
    minTimer.Stop();
}
```

## Enumerators (System.Collections.IEnumerator)

```
IEnumerator IEn = lstviewMessages.SelectedItems.GetEnumerator();
while (IEn.MoveNext())
{
    ListViewItem item = (ListViewItem) IEn.Current;
}
```

## Killing an entire application

```
Application.Exit();
```

## Sorting Ints, Strings, etc

```
ArrayList list = ...
list.Sort();
```

## Sorting—Custom Objects

```
public void sortMe()
{
    DateSort ds = new DateSort();
    myDateList.Sort(ds);
}

public class DateSort : IComparer
{
    public DateSort()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    public int Compare(object x, object y)
    {
        DateTime dx = (DateTime) x;
        DateTime dy = (DateTime) y;
        if (dx.Date == dy.Date)
        {
            return 0; //equal!
        }
        else if (dx.Date < dy.Date)
        {
            //list dx before dy
            return -1;
        }
        else
        {
            //list dx after dy
            return 1;
        }
    }

    public int CompareTo(object y)
    {
        return Compare(this, y);
    }
}
```

## Overriding ToString()

```
public class Message
{
    public override string ToString()
    {
        return "Subject is: " + this.Subject;
    }
}
```

## Checking type of object

```
object o = ...
if (o is int)
{
    int val = (int) o;
}
```

## Exception Handling

```
try
{
    //I hope things go ok...
}
catch (System.IO.IOException ioex)
{
    MessageBox.Show(ex.Message, "Disk stuff");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Uh oh...");
}
finally
{
    //clean up, close the file, etc
}
```

## Operator Overloading

```
public static Gayle operator +(Gayle m1, Gayle m2)
{
    //now we can call m1 + m2
    return new Gayle(m1.name, m2.name);
}
```

## Enumerations

```
public enum Justification
{
    Left,
    Right,
    Center
};

public bool isLeft()
{
    if (this.alignment == Justification.Left)
    {
        return true;
    }
    return false;
}
```

## Get and Set

```
public static int MinimumAge
{
    get
    {
        return minimum_age;
    }
    set //implicitly called when on "MinimumAge = 10"
    {
        if(value < 0)
        {
            minimum_age = 0;
        }
        else
        {
            minimum_age = value;
        }
    }
}
```

## Calling SQL-Getting Table

```

DataTable datatable = new DataTable();
string q = "select * from Blogs";
SqlDataAdapter adaptSQL = new SqlDataAdapter(q,
BlinkySQL);
adaptSQL.Fill(datatable);

foreach (DataRow drow in table.Rows)
{
    string commentbody = (string) drow["Body"];
}
    
```

## Calling SQL-Getting Single Value

```

string q = "select max(title) from Blogs";
SqlCommand command = new SqlCommand(q, BlinkySQL);
myCommand.Connection.Open();
string title = command.ExecuteScalar();
command.Close();
    
```

## Calling SQL-Execute Non-Query

```

string q = "delete from Blogs";
SqlCommand command = new SqlCommand(q, BlinkySQL);
BlinkySQL.Connection.Open();
command.ExecuteNonQuery();
BlinkySQL.Connection.Close();
    
```

## Webservice

```

edu.upenn.cis.wren.Blinky myBlinkySvc = new
edu.upenn.cis.wren.Blinky();

/* user exists in system */
if (myBlinkySvc.UserExists(this.textBoxName.Text))
{
    /* successfully logged in */
    if (myBlinkySvc.PasswordOK(username, password))
    {
        ...
    }
}
    
```

## SQL-Insert

```

insert into Blogs (UserID, BlogTitle)
values (1234, 'Yo')
    
```

## SQL-Delete

```

delete
from Blogs
where BlogID = 37515961
    
```

## SQL-Select

```

select UserID as 'User', max(datePosted)
from Blogs
where UserID <> 'Pat'
group by UserID
    
```

## Delegates

```

delegate void Fruitilicious(int num);
Fruitilicious yummy;

private void something()
{
    yummy = new Fruiticious(this.Apples);
    yummy(5);
}

private void Apples(int num)
{
    ...
}
    
```

## Threading

```

public delegate void Scruptious(string snacks);
Thread myThread;

private void startMyThread()
{
    roses = new Scruptious(this.Airport);
    ThreadStart marge = new ThreadStart(Bubbles);
    myThread = new Thread(marge);
    myThread.Start();
}

private void Airport(string word, int number)
{
    //update the UI, or do what you want.
    //This happens in the main UI thread
}

private void Bubbles()
{
    object[] parameters = {"Apples"};
    this.Invoke(this.roses, parameters);
}

private void StopMyThread()
{
    myThread.Abort();
}
    
```

## SQL-Update

```

update Blogs
set BlogTitle = 'hello'
where BlogTitle like '%hi%'
    
```

## XML-Reading

```
public void Open()
{
    XmlTextReader f = null;
    try
    {
        f = new XmlTextReader(path);
        while (f.Read())
        {
            if (f.NodeType == XmlNodeType.Element)
            {
                if (f.LocalName.Equals("Title"))
                {
                    string s = f.ReadString();
                    string id = f.GetAttribute("id");
                }
            }
        }
        note.Close();
    }
    catch (Exception e)
    {
        ...
    }
    finally

```

## File IO-Reading

```
if (File.Exists(path))
{
    System.IO.StreamReader file = null;
    try
    {
        file = new StreamReader(path);
        string data = file.ReadToEnd();
        file.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
    finally
    {
        if (file != null)
        {
            file.Close();
        }
    }
}
```

## XML-Writing

```
public void Save()
{
    XmlTextWriter f = null;
    try
    {
        f = new XmlTextWriter(m_path, null);

        f.Formatting = Formatting.Indented;
        f.Indentation = 6;
        f.Namespaces = false;

        f.WriteStartDocument();

        f.WriteStartElement("", "Note", "");

        f.WriteStartElement("", "Title", "");

        f.WriteString(this.m_title);

        f.WriteStartAttribute("", "id", "");
        f.WriteString(v.ToString());
        f.WriteEndAttribute();

        f.WriteEndElement();

        f.WriteEndElement();
        f.Flush();
    }
    catch (Exception e)
    {
        ...
    }
    finally
    {
        if (f != null)
        {
            f.Close();
        }
    }
}
```

## File IO-Writing

```
StreamWriter f = new StreamWriter(pth, false);
try
{
    f.WriteLine("stuff to write to file");
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
finally
{
    if (f != null)
    {
        f.Close();
    }
}
```

## Registry-Reading

```
RegistryKey regKey = Registry.LocalMachine;
regKey = Registry.LocalMachine;
regKey = reg-
Key.CreateSubKey("Software\\Glaak\\Fetcha");
object o = regKey.GetValue("lastUser");
if ((o != null) && (o is string))
{
    string name = (string) o;
}
```

## Registry-Writing

```
RegistryKey regKey = Registry.LocalMachine;
regKey = Registry.LocalMachine;
regKey = reg-
Key.CreateSubKey("Software\\Glaak\\Fetcha");
regKey.SetValue("lastUser", name);
```