1ST Class – Monday, January 06, 2014

Framework .NET Common libraries C# needs no ole DB

MS Visual Studio

C# - Console Application

Tools – Options – Text Editor – All Languages – General – Line Numbers

using System.Collections.Generic;

System.Collections. *(examples)* ArrayList IEnumerable (Interface)

Set, map, dictionary {} a group, or a sub-namespace <> List, linked-list, hash-set

Object.Equals

Object.ReferenceEquals

Pascal notation: Test (methods have initial caps) Register types of events will event. In that area.

Solution Explorer

Different languages can use the same resource files

partial – rapid application development – GUI controls

this.btnGenerate.Click += new System.EventHandler(this.btnCopy\_Click);

properties – lightning bolt – Events that are changed by this control.

structs - Essentially an object (not a primitive type), access to methods e.g. tryParse

numericupdown

MSDN C# keywords – const (like Final) Inclusive/exclusive values Method chaining

System.Windows.Forms Namespace

ListBox Class

Model View View Model (MVVM)

List as object in it and you can store the state (data in the object)

C# has is great for having a Class for describing the smallest thing ObjectCollection

[ListBox Class](http://msdn.microsoft.com/en-us/library/System.Windows.Forms.ListBox(v=vs.110).aspx)

[**Items Property**](http://msdn.microsoft.com/en-us/library/system.windows.forms.listbox.items(v=vs.110).aspx)

Smart-tag .. to setup data-bound items

[System.Windows.Forms](http://msdn.microsoft.com/en-us/library/System.Windows.Forms(v=vs.110).aspx)

[**ListBox.ObjectCollection Class**](http://msdn.microsoft.com/en-us/library/System.Windows.Forms.ListBox.ObjectCollection(v=vs.110).aspx)

White Hat hackers (find bugs) – 1st contribute to open projects, concentrate on a platform

Php, social engineering, sql injection

MS’s SQL: mssql (DBMS)

Bookstores – dead

eBooks – Alive

2nd Class – Monday, January 06, 2014

Object – Cards (flash cards, playing cards, keypunch cards, ) they have in common:

1. You can compare them, to organize them.
2. In C#, use IComparable (an interface)   
   <http://msdn.microsoft.com/en-us/library/system.icomparable(v=vs.110).aspx>
3. CompareTo (method) is an ANNOUNCEMENT and a CONTRACT.
4. [CompareTo(Object)](http://msdn.microsoft.com/en-us/library/system.icomparable.compareto(v=vs.110).aspx), that indicates whether the position of the current instance in the sort order is before, after, or the same as a second object of the same type.

#### Return Value

Type: [System.Int32](http://msdn.microsoft.com/en-us/library/system.int32(v=vs.110).aspx)  
A value that indicates the relative order of the objects being compared. The return value has these meanings:

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| Less than zero | This instance precedes *obj* in the sort order. |
| Zero | This instance occurs in the same position in the sort order as *obj*. |
| Greater than zero | This instance follows *obj* in the sort order. |

Natural order. Compare each piece of data until finished.

MS Visual Studio

Class Library – Card Class

A project for creating a C# class library (.dll)

“Value” is a free property (keyword) because this is the only property.

An optimization distinct to C#: a string comparisons are linear. C# as keywords…

“as” will type cast and assign the value.

“Is” will compare and return a Boolean.

Problem: In java, methods are virtual method (vm).

Parents have a vm and the child has precedence and is the one that runs.

C# does not have a vm so the parent is not replaced by the child. So we use “override”

And in addition, in the parent we use “virtual”

Abstract assumes it’s virtual.

Create a new project – console application

Open solution explorer – References – browse tab – select dll



3rd Class – Monday, January 13, 2014

Pitch Factor

Last week - Storing values into the controls themselves

Database – Tables – columns – Records (datafields) State portion of ADT.

Want ADT to share/store STATE with the app

Calling behavior of functions.

1. What does the GUI look like, how to draw ADTs
2. Start having a GUI and an object. Subject: Entity Name: \_\_\_\_\_

Example:

Routinely we write a

toString

Equals

IComparable (optional)

GetHashCode (optional)

Hashmap – make an index to an array (modulus of the length of the array)

To check (with an integer) and check if the object exists.

You have to have the whole set.

Wasteful (have a lot of RAM), extremely fast,

not always reliable, can have collisions.

Useful in storing a session.

Timer start or button, textbox

ClassTemplate

ClassName

ClassModifier

List<Field> “Has-a” – composition

DataTypes

Comparable: bool

HashCode:bool

Field

Name

AccessModifier

DataType

Enum {public, Protected,..}

4th Class – Wednesday, January 15, 2014

ContentmenuStrip StatusStrip ToolStrip fillList()

FlowLayoutPanel is ugly.

grp – GroupBox gives you a label with the box and controls radio button groups too.

Streams:

Source: external device:-keyboard

Destination: your program

Source: file

Destination: your program

Console.WriteLine (“ {0}”, ) // placeholders with param arguments

{0} class {1}

// State

// Propeties

// IComparable

// toString()

Pipes are good delimiters

ADTs.cs ClassCreationTool

Pg 20 should be the output

File Manipulation we have a java static utility class system.io.file

Delayed Writes – Tells the CPU that it can wait to a relative un-busy time to do writes to the harddrive. But could be a bummer if the PC crashes. The CPU needs to switch between threads, share the work. The cost advantages is saved state, advance time to operate instructions of instruction count, load state,

Page 524 – In the box see a file stream created.

Filemode enumeration – Append, Create, CreateNew, Open, OpenOrCreate, Truncate.

FileAccess enumeration – Read, ReadWrite, Write

Page 517 – Create text files and create binary files. Truncate will maintain the pointer to the file.

C# does not require the use of a try-catch block. Java does require it.

Try-catch-finally (same idiom as in Java)

FileStream fs= null;

try { // Connected to a file

// Connected to a database

// Connected to a remote system

} catch() {

} finally {

// Absolutely got to get done

// Close the file

// Close the database

// Close the remote system

}

5th Class – Monday, January 20, 2014

Common Language Interface (CLI) loads Class Definitions onto a stack frame.

A call we get us to leap to the heap.

And objects instantiated are put in the heap .

To call a method or use a property (e.g. .p) it is holding an address where we start to read the name of a class that refers to the CLI class definition. And instructions are4 loaded and ran on

p.FirstName = “Jason”;

**static** Utility classes has shared methods that are static because you do not have to instantiate an object for the method. There is only one. It does not store any state so it doesn’t need to store data in the heap.

**Override** keyword to override the method.

**virtual** compareTo the parent. Then override it for the child. write virtual table lookup

**new**

protected (Java) – children classes, other classes in package

protected (C#) – splits it into two separate access modifiers:

protected (C#) – children classes

internal – other classes in package

6th Class – Wednesday, January 22, 2014

Conditional statements – switch (Integral – char, int) Always have a Default case!

In Visual C, string data can be used – But bad coding style.

You can use enums (Monday, Tuesday, Wednesday) integral type.

In C#, does not allow fall-through.

In C#, a switch can have multiple labels. Note – A label is not a case without a body.

case “A”: // not a case but only an additional label for case B.

case “B”: // a case

body; // doesn’t fall-through to case C

case “C”:

default:

break;

Arrays – A compound data type of one type but many elements of that type.

// pointer of [offset=0] goes to memory address of the first element in the array

int[0] numbers; We name the array. We can’t grow (or shrink) the size of the array.

Managed Memory – now we are protected somewhat by the compiler (out of bounds)

Parrallel arrays – Use the index of one array to get the data at the index of another array.

Multi-dimensional arrays – C# uses brackets for each dimension.

Example: Table:

int [][] Students [] []

// The problem is the Array uses only one type to advance the offsets in memory

Not the solution. So the (int) is a limited data type. And converting numbers to a (string) is tedious.

Syntax advantage – write array: int[] nus = {1,2,3,4} Or in Javascript we need totype:

Num[0] = 1;

Num[1] = 2;

Num[2] = 3;

Num[3] = 4;

Array can hold any object: Example: Textbox[]

Enums are for the conditional statements (it protects the programmer)

Arrays are output for the user, Example: A class

Dog[] dogs {new Dog(“Spot”,”Dalmatio”),

new Dog(“Maks”,”Terrier”),

new Dog(“Jack”,”Pittbull”),

new Dog(“Sam”,”GermanSheppard”) };

dogList.AddRange(dogs);

Data can grow and shrink – use a list (complex) RISC vs CISC. Example: A class Employee

A class EmployeeLoader to fill employee data

Employee[] employees = new Employee[10];

EmployeeLoader.loadEmployees(employees);

listBox1.Items.AddRange(employees);

call stack is on the heap.

keyword “new” 1) reserves space, 2) calls a constructor, 3) returns the reference

The parameter of an array is (array[] name) reference-based data management

7th Class – Thursday, January 22, 2014

GUI Interface Desugn:

Good books:

“Don’t Make Me Think”

“The Design of Everyday Things” (web-based design)

“Websites that Suck” (web-based design)

“25 Worst Websites in 2013” (web-based design) www.reforms.net

Good ideas in creating objects:

CurrentMode<Mode> Mode – enum Mode {Add, Edit, View}

ContactList – List<Contacts> Contacts – {Name, Address, Phone, Email}

Design Database by writing a Class Diagrams

Normalize Tables –

Customer – firstname, LastName, Address, Phone, Email

Order – List<OrderDetail> , Customer, Subtotal, Tax, Total, Date

OrderDetail – Qty, Book, Subtotal, FK:Order, FK:Book

Book – Title, Author, ISBN Look at the receipt:

High Cohesion – in the objects

Low coupling – between them

TransactionNumber = DateTime-MACaddress-SequentialNumber

cboTitle.Items.AddRange(BookData.ReadBookData().ToArray());

cboTitle.SelectedIndex = 0;

8th Class – Monday, January 27, 2014

int[3][3][3] Multi-dimensional array contiguous array, symmetrical

int[3,3,3] Jagged array – array of arrays

Webclass – Week 4 – Cars.dat file

(Pipe separated)

Vehicle

licenseNumber : String

make : String or an Enum

model : String or an Enum

year : int

color : String or an Enum

Owners : List<Person>(many-many)

-------------

Person

Vehicles : List<vehicle>(many-many)

Warrants: List<warrant>(many-many)

Firstname: string

Lastname: string

Address: string

IdentifyMarks: string

Driver

Warrant

(zero or more)

Predicate<LT> Delegate (COM 271)

9th Class – Wednesday, January 29, 2014

Webclass – Week 3 – PassArray.zip file

**CryptTool**

**Test connection to MS SQL Server – database**

---- Personal information for Michael Fetick ----

Student number db84270 Microsoft SQL Server account information - PRIVATE

Hello Michael,

Here is your information for the Microsoft SQL Programming Course:

Your username:          db84270

Your password:          5a0c1de0

Your database name:     db84270

MSSQL Server FQDN:      mssql-2-34.int.coleman.edu

or at this IP address:  172.16.2.34

Have Fun!

School: MS Visual Studio – View – Server Explorer – Data Connection (Right-Click) – Add Connection… –

Home: MS Visual Studio – View – Server Explorer – Data Connection (Right-Click) – Add Connection… –

Microsoft SQL Server – Add Connection (Form). Server Name: (Nothing listed)

(Have installed) MySQL – MySQL Workbench 6.0 CE. [Started]

(Have MySQL Connections already set up)

mine – pm84270, pm84270, pm84270@linuxsandbox.coleman.edu:22

atHomeDefault – 127.0.0.1, 3306, root, Test Connections (Parameters are correct)

[Configure Server Management] – Configure Local Management:

(Also have installed) MS SQL Server 2008.

Data provide

Dsn name ms-sql-2-34  
Connection String - Data Source=mssql-2-34;Initial Catalog=db84270;Persist Security Info=True;User ID=db84270;Password=\*\*\*\*\*\*\*\*\*\*\*

In the VS, it uses the ‘MS SQL Compact Edition’

- Local Database

Entity Data Model – Empty Model

**Ado.Net**

Database flights

Building an auto-incremented primary key

Come back later to right-click – table definitions,

and Show Table Data

where filter

Char uses ascii , 8-bit

**Nchar** uses Unicode unsigned, 16-bit

Faster to compare because they are linear in width

Vchar, Char – is for speed

nchar – saving space

Number of chars actually stored, spaces that are padded.

Nchar 50

small money

small int

Binding Navigator (GUI arrow controls)

Binding Source – Properties – Text

(**DOI**) Digital Object Identifier – to uniquely identify an entity

Database design – read by many, edited by one

Edit mode locks out others

View -

Every table is a **B-Tree** for binary search

The ‘Back Tick’ is the “ ` ” (left of the 1 key)

The ‘Front Tick’ is the “ ‘ ” (left of the Enter key)

**TSQL** – Transact SQL

sqlisam does NOT support transactions

Innodb does support transactions

Connected Architecture – Persistent connection between a DBMS and our app but is using resources.

Dirty data – different state from the source

Disconnected Architecture – We cut the connection.

How do we handle obsolete data..

Keep the Disconnected Architecture but after a few seconds…

Dump your data and reload (pull) your original data again. Into your cache.

Current critical data - Stock Prices

**Lossy**

**lossless**

Data Adaptor – complex, bloated, mimics the DBMS locally.

Data Set– complex, bloated, mimics the DBMS locally.

**SQL auto-increment**

IDENTITY (1, 1) // Seed in the increment

10th Class – Monday, February 3, 2014

Professor Jason Abel was not present. Robert substituted and we had lab time.

11th Class – Wednesday, February 5, 2014

Professor is ~~struggling~~ researching with “APA Format.”

IDE, C#, Class, Cntrol Stmt, Interface Controls, Files, Streams

Library – .NET is the standard library of classes

We reference the library, go to class when we call it, it goes to the heap, gets the object.

Standalone library, dynamic link library. We create links to it when we run.

In C++ we had static libraries that loaded in compile.

Class = Type of objects = DLL (Dynamic Link Library)

w/ Method area. i.e. object1.do() we find the do in the class definition

this – refers to the pointer of the object

Stack – Frame has local variables

Heap - Objects in the heap (state, it persist)

Ref Table, w/ Ref Pointers,

Memory-CPU – Where classes loaded 1.5gb reserved for devices, now at 64bit we get 4TB ram limit

Reads rom at pointer zero (read only).

Main’s address is some distance from zero.

It’s more like dynamic program generation.

Streams: USB driver, HHD driver, SSD driver, Network

Directory – files addresses, id number

Files – data ray written to a persistent format

NIO – static class

<<ISerializabgle>>

SerializationInfo – getObjectData()

Object to CSV (looping, parsing, instantiation, platform agnostic)

Object to Binary (platform-specific)

Object to Registry (going away from that)

[NonSerializable]

Try {

// external access to do anything

} catch (exception e) {

// handle errors

} finally {

// release resources, close stuff

}

Target the try-catch blocks only when necessary because it uses memory

- - -

Asychronize encryption

Google’s page relevance algorithm

(Marissa Meyer, freaky laugh)

Linked-list versus an array

* Linked-list does not support random access. You can easily insert and delete items. Good for a stack because there is no size.
* Array supports random access. You can’t easily insert and delete items. Bad for a stack because there is a fixed size and may run out of room.

Hashmap – Have an array with each element pointing to an linked list.

ADO.NET

Download .NET modules (VB.NET, C#.NET…)

Linux

Parser – compiler – Linker

Parsers(C#, VB, J#, F#)

Data provider is an implementation of abstract classes in the Oracle client.

.NET Framework Data Providers

Calls a DataReader, DataAdapter,

Two Options:

- Go little - Connected Architecture (Ties-up external resources)

- Go big - Connected Architecture (Ties-up internal resources, i.e. ram)

Connected Architecture

- Connection

- Command

- Data reader (Queries only)

Disconnected Architecture (resource hog on your local system) (Batman architecture)

- DataSet (memory representation of data)

DataTable

DataTable

- DataAdapter (Designed for a specific data table)

Connection

Command 1 – Insert

Command 2 – Update

Command 3 – Delete

Command 4 – Select – Data Reader

DataReader (To only read data)

DataSet (To cache the data)

LINQ to DataSet

DataSets, DataTables, and DataViews

ff647768.aspx

Employees (empId, last, first, title)

DataEmployees (empId, last, first, title)

popDataSet // Creating an object that takes on the properties of a database table

(takes a ton of memory – it’s a big object)

For a Dataset we need 1) connection, 2) sql statement

Dataset is empty, we will fill it with data.

DataTable dt = ds.Tables[“products”];

dt. //Select, AcceptChanges, AsDataView (LINQ-enabled)

dt.Select // Allows you to use the SQL clause “Sort”, “Order By”

To bind, add a project data source, i.e. a database

Dataset (fill,getData), bindingSource, TableAdapter (:Flight TableAdapter, :Passenger TableAdapter) (fill,getData) (Wizard)

AT – placeholder

AND – parameterized query

Using raw SQL is a bad idea.

First setup a reference (so it can refer to its peer):

Obj2 = new Form(this);

Odj1 = otherForm

Do ()

{

obj2.show();

obj1.hide();

}

Register

forms List[…] // {Statics are underlined}

For now, get your data into a database.

12th Class – Thursday, February 6, 2014

Who: CSR

1. Add – Enables the form
2. Enter relevant info
3. Create customer
   1. Validate info
   2. Confirm info
   3. If rule fail, go to step 1.
4. Add customer to record
5. Update database
   1. If database is not available, log the record
   2. Attempt later to add record
6. End

Who: CSR

Pre-condition: Record must exist

1. Select customer
2. Populate customer data in the form
3. Enable editing via Edit button, save enabled
4. Input customer data
5. Validate customer data
   1. If invalid customer data, go to 4.
6. Create customer data read, commit to database
7. End

Customer – Order

Credit card –

Address –

Email –

Order

Customer

Add Title

Update Quantity

Delete Title

Cancel Order

MVVM – MS model-view-view-model

Affordance – design an interface that assists the interface by the natural way to do things.

Core components –

System.Data.SQLClient;

cmd = …

cmd.Parameters.AddWithValue(“@lastname”, input);

Always use Parameterized queries

- problem with concatenation – I don’t know what is in {input}

E.g. ”Select \* from Employee WHERE UserId = ‘” + input + “’”;

SQL Injections

input = “0’ or ‘A’ = ‘A’; drop Employees”

Library “Pear” – PQDBI

Set in a loop  
{

Insert into Table (first, last, mi)

Values (@first, @last, @mi)”

Values (@book.Title, (@book.Isbn, @book.Qty)”

}

“Delete from Table WHERE parm = @parm”

13th Class – Monday, February 10, 2014

DataGridView Tool – A control to be bound to a data source.

A great tool without much purpose, it is pretty.

We use the Adapter to commit the changes.

Look at the events – i.e. CancelRowEdit, ControlEdit RowChanged, RowLeave.

Right-Click, selecting the Binding Source gives us the CustomerDataAdapter and other stuff

Look at what CustomerDataAdapter has for us,

On RowLeave,

There is an edit mode and a commit mode.

We can add a button calling it “Commit” and

“CommitEdit”

“EndEdit”

How to bind data to the window…”

“Walkthrough validating data…”

Fyi – ClearBeforeFill will check if you already filled data, with it checking for collisions.

To retrieve a row, you don’t get a list, you get a DataGridViewRowCollection

DataGridView – (a doubly linked list)

Rows - DataGridViewRowCollection – there are Columns, a DataGridViewColumnsCollection

Item - DataGridViewRow – … a DataGridViewColumn

Cells - DataGridViewCellCollection –

Item - DataGridViewCell – (String, Column Name, Ordial Index)

Tool-tips,

Inefficient to

It is better to maintain an order with order details, recalculate the

DataGridView supports

- IList

- IListSource

- IBindingList

- IBindingListView

Use a list, manage the columns. Or use a DataGridView and get sorting on column.

Can use binding to a BindingSource.

Inequality for all, Robert rice

Dan Bennit, “The Illusion of Consciousness”

A Delegate is called “The Adapter Pattern”

The IDisposal Interface – clears an object from memory. Unmanaged resources/code.

Managed code is on the virtual machine (VM).

The database is outside your VM it uses a driver and is native.

But a Dataset is managed code.

Using statement, important for reader, etc…; to safely dispose of unmanaged code.

It calls the correct dispose method to correctly unload the objects.

A handle is a pointer to data that is unmanaged.

Component is a managed resource that represents the unmanaged data on the inside.

Handle = IntPtr.Zero;

// In C++ you have to set the value of every pointer to zero, when you are done using it.

Unmanaged code does not have tools to dispose them so we have “Using.”

using(SqlCeDataAdapter a = new SqlCeDataAdapter(

“Select \* FROM Animals”, c))

{

// Use DataAdapter to fill DataTable

DataTable t = new DataTable();

a.Fill(t);

}

UtilityClassDB

AddCustomer(Customer)

GetCustomer(): Customer

SaveOrder(order)

GetBook(.):Book // using a Dataset and DataAdapter

GetConnection(Dictionary):Connection

# Dictionary<TKey, TValue> Class (A hashmap to hash the key) Store a key in … set(key, value) Uses a set of predefined keys.

Choices - Versus having it on the other side, at the database.

At database

Stored Procedures

Retrieving Queries

Updating Records

Link table: PassengerList

Try to add a new passenger in PassengerList (it has constraints)

i.e. without a linked passenger record then the query fails so we don’t get bad data in our database.

CREATE TABLE Customer(CustId int NOT NULL, LastName varchar(20) NOT NULL, FirstName varchar(15) NOT NULL, Address varchar(30)

DEFAULT NULL, City varchar(20) DEFAULT NULL, State char(2) DEFAULT NULL, Zip varchar(10) DEFAULT NULL, Phone char(12) DEFAULT NULL, Email varchar(35)

DEFAULT NULL, CardType varchar(16) DEFAULT NULL, CardNumber varchar(16) DEFAULT NULL, CardExpire varchar(5) DEFAULT NULL, PRIMARY KEY (CustId));

INSERT INTO Customer

(LastName, FirstName)

VALUES ('Richardson', 'John')

INSERT INTO Customer

(LastName, FirstName, Address, City, State, Zip, Phone, Email, CardType, CardNumber, CardExpire)

VALUES

('Richardson', 'John', '1524 Center Dr.', 'La Mesa', 'CA', '91945', '619-358-6578', 'jrich@cox.net', 'Visa', '4224369865742591', '03/07'),

('Lolan', 'Bobbi', '122 Elysian Fields', 'New Orleans', 'LA', '73401', '704-987-6421', 'blolan@no.rr.net', 'Master Card', '5623326598754128', '02/05'),

('Smith', 'Tim', '12597 1st Ave.', 'Chula Vista', 'CA', '92174', '619-658-8794', 'tsmith@sbcglobal.net', 'Discover', '6598234568710954', '04/06'),

('Edwards', 'Mitch', '19872 Pine Court', 'Kansas City', 'MO', '69877', '648-987-0938', 'mitch@earthnet.com', 'American Express', '370044875669745', '10/04'),

('Brown', 'Edward', '14598 Poway Road', 'Poway', 'CA', '92064', '858-486-6897', 'ebrown@hobo.net', 'Master Card', '5236698756824712', '09/05'),

('Rowland', 'Mike', '32657 Jasmine Place', 'Park City', 'UT', '86987', '236-568-9874', 'roland@hotmail.com', 'Visa', '4225639856875214', '12/04'),

('Perkins', 'Anthony', '235 Circle Dr.', 'La Mesa', 'CA', '91942', '247-986-3274', 'psycho@bates.com', 'Master Card', '5668213659875897', '12/05'),

('Jones', 'Ralph', '2 Shady Lane', 'Chula Vista', 'CA', '92105', '254-987-2327', 'shady@nightmare.com', 'Discover', '6987986215476387', '01/07');

Storing an image in an array: Blobs, clobs, store the path to an image.

14th Class – Wednesday, February 12, 2014

MaximumPC

Parameters, commands, connections.

15th Class – Wednesday, February 19, 2014

Next week we’ll look at custom handlers with Delegates.

Stored procedures are not supported by a local database.

2013 MS SQL Server Express SP3 can do it.

Functions, Triggers, Assemblies.

1. Project status
2. 2. Stored procedures

CONNECTED ARCHITECTURE - LIVE CONNECTIONS (STAYS OPEN)

SQL Connection

SQL Command. Command text = “” // Only needed during the select

SQL Reader // Local interface to the set of methods available on the client side

Parameterized Query has place holders in the Command.text within the “string.”

It has the placeholder name and its value

Standard query, create command, get data back

1. Parsing the SQL
2. Marshalling an action plan
3. Execute

Parametrized query requires the same:

1. Parsing the SQL // Only do this once
2. Marshalling an action plan // Only do this once
3. Execute

DISCONNECTED ARCHITECTURE - LIVE CONNECTIONS (STAYS OPEN)

Stored Procedure – A function of the management system when it gets called.   
A lot of correlation to a set of methods

The Command.Type is changed to a Stored Procedure

Options – Text Editor – Fonts – Plan Text

Don’t use the “sp\_” prefix because it will look through the system database before it looks for yours.

CREATE PROCEDURE CustOrdersDetail

@OrderInD int

AS

SELECT ProductName,

UnitPrice=ROUND(Od.UnitPrice, 2),

Quantity,

Discount=CONVERT(int, Discount \* 100),

Extended Price…

FROMProducts P, [Order Details] Od

WHERE Od.ProductId = P.ProductID and Od.OrderID = @OrderID

Input Parameter: // We can put data into a method body. MethodCall(input Param)

Return Type Parameter.

Unusual (Done only in C#) It returns TWO THINGS (e.g. return a bool and return an int) (Weird One)  
 bool Int tryParse(1, out num) { return bool}  
 has an input parm and an output parm.

Knights Tour – Is it possible for the knight chess-piece to move around the board and traverse every square on the board without duplicate?

Class Knight {

int x, y; // out parms

move (x, y);

int getPosition (out x, out y)

tryMove(x, y)

}

You don’t need a heavy DataReader…

Command.Type = Command.Type.StoredProcedure;

Command.Text = “sp\_Orders\_By\_EmployeeId2”;

// create input parameter (a reference)

// create output parameter (a reference)

// create return value parameter (a reference)

// execute reader

// Process the result set and close reader

// Display output parameter value

// Display return value // keyword: “return value”

Create procedure sp\_Orders\_By\_EmployeeId2

select @ordercount = cont

from orders

where employeeid= @employeeid

return

/\* TO CONNECT AT SCHOOL - COLEMAN UNIVERSITY

\*/

builder.DataSource = "mssql-2-34";

builder.UserID = "db84270";

builder.InitialCatalog = "db84270";

builder.Password = "5a0c1de0";

//

return new SqlConnection(builder.ConnectionString);

Design Patterns //Frameworks for writing code – promote high-cohesion, low-coupling

- MVC

- MVVM

- Singleton

- Observer Pattern. // Upon which data binding is built.

- Factories

When you have a resource, you make it, as an example:

Singleton Pattern:

private final Object o;

public Object getO()

{

If (o == null)

{

O = newOject();

}

Return o;

}

Observer Pattern. // Upon which data binding is built.

<http://msdn.microsoft.com/en-us/library/ee817669.aspx>

<http://msdn.microsoft.com/en-us/library/ms752347%28v=vs.110%29.aspx>

News Station with… Sensors: Downtown, La Jolla, Imperial Beach,

Object – List<Observers> observers

Observer – list<enums> Coronado, 1B (push polling) – **Notify()** //contains its own state

The observers handle their own business.

- This has to do with data binding

because the data binding has a data source that is handled the binding manager

the binding manager notifies all the list<controls> that the state has changed.

This is common with record change, moving through records.

private final Object o;

public Object getO()

{

while (more obs)

{

Obs.notify() // changed

}

Return o;

}

customerBindSource.position = 5;

bindingNavigator…

16th Class – Monday, February 24, 2014

Book lookup service = order, commit

Customer lookup = order, commit

Tables

Project 5 has at least:

- Standard queries

- Parameterized queries

- Stored procedures (Order)

Library ADTs (customer, book)

Data Management Classes (utilities)

Inheritance

OOP

DB Interactions

GUI

Program Flow

C# Concepts

- - -

Delegates, Com 275

OSI Model (7 layers)

-----

TCP Protocol stack

Transport – Transmission Guaranteed

Network – IP Addr

Data-Link – MAC Addr

Physical

----

Behind Gateways – Routers - Subnets

Network Address Translation (NAT)

Time to lIve – TTL

Invalid IP Address

192.168.0.1

10.0.0.1 (Mac)

Datagrams – UDP

Berkley socket, AppleTalk (IPXSPX more efficient)… Compatible to many protocols,

Socket: Connection

Streams

Open()

Close()

Source.write()

Destination.read()

Server

To make a basic server:

A process on a machine that listens for request

From a chine on a port, IP:Port

80 HTTP

443 HTTPS

20 FTP

21 FTP

25 SMTP

Don’t do a ‘port scan’ on anyone.

User Datagram Protocol (UDP)

TCP – Reliable

UDP – Unreliable, but fast. No handshake.

Make another thread to monitor a socket. Threads have to talk to each other. Make coupling work.

Socket Class…

Low level class. Tell it how much, buffer reading,…

Composite relationship, other classes contain sockets.

Tcpclient…

Send the data with a handshake.

NetworkStream…

Qulacomm vuforia

Deloper. Vuforia.com

Blurr computer vision

Project 5 debug:

// Utility method to create the order

public StringBuilder BuildOrder(StringBuilder receipt)

{

// Add the customer data

RegisterCustomer(order);

// Add the items the customer has put on the order (cart)

ItemizeOrder(order);

// Display the order details for confirmation and a receipt

sb = DisplayOrder(order);

return sb;

}

FIX THIS TO USE THE DATASET

// Utility method to register the Customer in the Contact List

protected void RegisterCustomer(OrderDetails order)

{

// Assign the (GUI) user-input to the properties of the Customer

order.Customer.FirstName = txtFirstName.Text;

order.Customer.LastName = txtLastName.Text;

order.Customer.Address = txtAddress.Text;

order.Customer.City = txtCity.Text;

order.Customer.State = cboState.Text;

order.Customer.ZipCode = txtZipCode.Text;

order.Customer.Email = txtEmail.Text;

// Add the Customer to the List of contacts

List<ContactDetails> contactList = new List<ContactDetails>();

contactList.Add(order.Customer);

contactList.Sort();

}

// Utility method to display the order to the customer

protected StringBuilder DisplayOrder(OrderDetails order)

{

/\*

\* Specified in the Student Handbook on page 20.

\*/

// StringBuilder appends to one string vs many literal strings;

// delimiter "|" is inserted to parse the lines and items later,

// from the single line of each order.

//

// Start the string with the order date and the customer data

sb.AppendFormat("Order Date: {0}|{1} {2}|{3}|{4}, {5} {6}|",

order.OrderDate,

order.Customer.FirstName.ToString(),

order.Customer.LastName.ToString(),

order.Customer.Address.ToString(),

order.Customer.City.ToString(),

order.Customer.State.ToString(),

order.Customer.ZipCode.ToString()

);

//order.Customer.Email.ToString()

// Append the list of books ordered

sb.AppendFormat("{0,-24}\t{1,-5} {2} ({3}) {4:-c2,15}|",

"Title", "Author", "ISBN", "Quantity", "Price");

foreach (ItemDetails item in order.ItemList)

{

sb.AppendFormat("{0,-24}\t{1,-5} {2} ({3}) {4:-c2,15}|",

item.Title, item.Author, item.Isbn, item.Qty, item.Price);

}

// Append the summary line

sb.AppendFormat("SubTotal: {0} Tax: {1} Shipping: {2} Total {3} |",

order.Subtotal, order.Taxes, order.Shipping, order.Total);

// Append the final line to separate orders

sb.AppendFormat("-----------------------------------------------------------------\n");

return sb;

}

// Utility method to write the order

public static void WriteOrder(StringBuilder sb)

{

// Pg528. Writing to a sequential access text file

// declare a delimiter, an output text file, and some arrays

const string FILENAME = "OrderData.txt";

if (File.Exists(FILENAME))

{

try

{

// Use the File.WriteLine method to write the StringBuilder line.

// It creates a FileStream object, associates it with a StreamWriter

// and later, closes the file.

// Write to order file

File.WriteAllText(FILENAME, sb.ToString());

}

catch (Exception)

{

// throws NotEmplemented

}

finally

{

// the File.WriteAllLines method also, closed the file.

}

}

}

private void cboCustomer\_SelectedIndexChanged(object sender, EventArgs e)

{

customer = (ContactDetails)cboCustomer.SelectedItem;

if (cboCustomer.SelectedIndex >= 0)

{

grpCustomer.Text = "CUSTOMER: {" + customer.CustId + "}";

txtFirstName.Text = customer.FirstName;

txtLastName.Text = customer.LastName;

txtAddress.Text = customer.Address;

txtCity.Text = customer.City;

cboState.Text = customer.State;

txtZipCode.Text = customer.ZipCode;

txtPhone.Text = customer.Phone;

txtEmail.Text = customer.Email;

cboCreditCard.Text = customer.CardType;

txtCcNumber.Text = customer.CardNumber;

nudCcExpMM.Value = Convert.ToDecimal(customer.CardExpire.Substring(0, 2));

nudCcExpYYYY.Value = Convert.ToDecimal("20" + customer.CardExpire.Substring(3, 2));

}

// Disable the controls for editing customer data

DisableCustomerEditing();

}

Use MS SQL Express

17th Class – Wednesday, February 26, 2014

<http://www.dotnetperls.com/datetime>

Screen capture software:

Steve – ScreenFlow, SoundFlower

Jason – Capture

JSON – JavaScript Object Notation

Obj { “name” = “value” }

JSON is used for Ajax rather than XML. It’s less provost.

Mongol – A type of database. Operates in JSON

AJAX – Cross-scripting restrictions is what is a challenge.

How to combine technologies

[Serialized] – An Attribute – This is metadata, information about the data.

Attributes are common in C#.

Do not serialize reference types, use: [NonSerialized]

The scope of [NonSerialized] is only just the next line.

Reflection – write code that analyzes other code.

Use composition when you can – an object contains the state of another.

Get away from relying on constructors and attributes.

Static Factory – A good choice when you don’t have values

Builder – A helper class

Has the private constructor NutritionFacts

Method chaining

Book, “Effective C#” - Good design processes.

Codinghorror.com

**John D. Carmack**

[](http://en.wikipedia.org/wiki/John_D._Carmack)[www.gemind.com.br](http://www.gemind.com.br)

John D. Carmack is an American game programmer and the co-founder of id Software. Carmack was the lead programmer of the id video games Commander Keen, Wolfenstein 3D, Doom, Quake, Rage and their sequels. Carmack is best known for his innovations in 3D graphics, and is also a rocketry enthusiast and the founder and lead engineer of Armadillo Aerospace. In August 2013, Carmack took the position of CTO at Oculus VR. [en.wikipedia.org](http://en.wikipedia.org/wiki/John_D._Carmack) · Text under [CC-BY-SA license](http://creativecommons.org/licenses/by-sa/3.0/)

VR

Omni-treadmill.

Glucose for battery technology for storing power.

Wally Facebook interaction.

CREATE TABLE [dbo].[Customer] (

[CustId] UNIQUEIDENTIFIER NOT NULL,

[LastName] VARCHAR (20) NOT NULL,

[FirstName] VARCHAR (15) NOT NULL,

[Address] VARCHAR (30) DEFAULT (NULL) NULL,

[City] VARCHAR (20) DEFAULT (NULL) NULL,

[State] VARCHAR (20) DEFAULT (NULL) NULL,

[Zip] VARCHAR (10) DEFAULT (NULL) NULL,

[Phone] VARCHAR (12) DEFAULT (NULL) NULL,

[Email] VARCHAR (35) DEFAULT (NULL) NULL,

[CardType] VARCHAR (16) DEFAULT (NULL) NULL,

[CardNumber] VARCHAR (16) DEFAULT (NULL) NULL,

[CardExpire] VARCHAR (5) DEFAULT (NULL) NULL,

PRIMARY KEY CLUSTERED ([CustId] ASC)

);

CREATE TABLE [dbo].[CustomerOrders] (

[CustomerOrderId] UNIQUEIDENTIFIER NOT NULL,

[DateOrdered] DATETIME2 (7) NOT NULL,

[Subtotal] MONEY NOT NULL,

[Taxes] MONEY NOT NULL,

[Shipping] MONEY NOT NULL,

[Total] MONEY NOT NULL,

[CustId] UNIQUEIDENTIFIER NOT NULL,

PRIMARY KEY CLUSTERED ([CustomerOrderId] ASC),

CONSTRAINT [FK\_CustId] FOREIGN KEY ([CustId]) REFERENCES [dbo].[Customer] ([CustId])

);

CREATE TABLE [dbo].[OrderDetails] (

[OrderId] UNIQUEIDENTIFIER NOT NULL,

[ISBN] NCHAR (15) NOT NULL,

[Title] NVARCHAR (50) NOT NULL,

[Author] NCHAR (30) NOT NULL,

[Price] MONEY NOT NULL,

[Qty] INT NOT NULL,

[CustomerOrderId] UNIQUEIDENTIFIER NOT NULL,

PRIMARY KEY CLUSTERED ([OrderId] ASC),

CONSTRAINT [FK\_CustomerOrderId] FOREIGN KEY ([CustomerOrderId]) REFERENCES [dbo].[CustomerOrders] ([CustomerOrderId])

);

At school:

Microsoft Visual Studio 2010 Ultimate

Version 10.0.40219.1 SP1Rel

18th Class – Monday, March 3, 2014

Encryption

Never store the key on the same server as the encrypted data.

Reduces the likeliness of being hacked.

Four classes of encryption

Advanced Encryption Standard (AES) Type 3 encryption

RijndaelManaged Class (Used in C#) The APIs:

<http://msdn.microsoft.com/en-us/library/system.security.cryptography.rijndaelmanaged(v=vs.110).aspx>

IV – Initialization Vector (Sides where it starts to create random characters)

The seed is the origin of the key. Seeds in Random numbers is the TimeDate stamp.

// Create the streams used for encryption.

using (MemoryStream msEncrypt = new MemoryStream())

{

using (CryptoStream csEncrypt = new CryptoStream(msEncrypt, encryptor, CryptoStreamMode.Write))

{

using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))

CryptoStream is a wrapper for the MemoryStream.

Hashing is one-way

Encryption is two-way

cipherText – the encrypted data

<http://msdn.microsoft.com/en-us/library/system.security.cryptography.rijndaelmanaged.generateiv(v=vs.110).aspx>

Symmetric Key encryption (agree on a key) Sharing a secret

Public Key – Private Key combination (anyone on the web)

Setup can be mysterious – read research

Hashing – (To challenge a person for identity without storing it)

Salt – random number generation at it’s finest.

<http://stackoverflow.com/questions/2138429/hash-and-salt-passwords-in-c-sharp>

<http://stackoverflow.com/questions/5252943/why-we-use-the-salt-to-secure-our-passwords>

<http://msdn.microsoft.com/en-us/library/system.security.cryptography.rfc2898derivebytes.salt(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/library/system.security.cryptography.passwordderivebytes.salt(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/magazine/cc163884.aspx>

<http://www.truecrypt.org/> (Free open-source on-the-fly encryption)

block cipher

<http://en.wikipedia.org/wiki/Block_cipher>

Rng – Random number generator RNGCryptoServiceProvider

<http://msdn.microsoft.com/en-us/library/system.security.cryptography.rngcryptoserviceprovider(v=vs.110).aspx>

Storing on SQL Server as a string with .ToBase64String(salt);

Linux has a Shadow password file, and code is tested against the hash. Line 122

SHA-2 – Secure Hash Algorithm – 256 bits

<http://en.wikipedia.org/wiki/SHA-2>

SHA512 Class – 512 bits

<http://msdn.microsoft.com/en-us/library/system.security.cryptography.sha512(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/library/s02tk69a(v=vs.110).aspx>

19th Class – Wednesday, March 5, 2014

COM objects and DLL for C++

C++ runs native

C# runs on a virtual machine so there is the performance drawback.

Delegates

Delegate -> Adapter pattern

JButton

Public void ActionPerformed (ActionEvent e)

C#, Java not first-order objects, can’t normally pass them around

<http://msdn.microsoft.com/en-us/library/ms173173(v=vs.90).aspx>

Windows 8 – Metro = new platform frontier for mobile app development

<http://msdn.microsoft.com/query/dev10.query?appId=Dev10IDEF1&l=EN-US&k=k(VS.AMBIENT);k(TargetFrameworkMoniker-%22.NETFRAMEWORK%2cVERSION%3dV4.0%22)&rd=true>

<http://msdn.microsoft.com/en-us/library/67ef8sbd(v=vs.100).aspx>

20th Class – Thursday, March 5, 2014

tablelayoutpanel

flowlayoutpanel

autosize

autosizemode

minimum size

anchor (top.left) for the most part

Dock – fill

Split container

Flow control

Magnification API Overview

<http://msdn.microsoft.com/en-us/library/windows/desktop/ms692402%28v=vs.85%29.aspx>

Book – **About Face: The Essentials of User Interface Design**

<http://www.amazon.com/About-Face-Essentials-Interface-Design/dp/1568843224>

21st Class – Monday, March 10, 2014

CRAP – Contrast, Repetition, Alignment, Proximity.

<http://www.presentationzen.com/chapter6_spread.pdf>

<http://blog.teamtreehouse.com/how-crap-is-your-site-design>

<https://comminfo.rutgers.edu/blogs/sgarwood/contrast-repetition-alignment-proximity.html>

<http://voices.yahoo.com/the-crap-design-principle-contrast-repetition-alignment-3710716.html>

<http://web.nmsu.edu/~jasheppa/courses/eng218/crap_principles.html>

Mono – Csharp Compiler – C# on a platform other than Microsoft

Ecma-334, C# Language

[www.Ecma-international.org](http://www.Ecma-international.org)

**Common Intermediate Language** (**CIL) – was** MSIL – Microsoft Intermediate Language

<http://en.wikipedia.org/wiki/Common_Intermediate_Language>

<http://msdn.microsoft.com/en-us/library/c5tkafs1%28v=vs.90%29.aspx>

WHQL

<http://en.wikipedia.org/wiki/WHQL_Testing>

<http://www.techpowerup.com/articles/other/26>

HWCL

Don’t always rely on the Microsoft Drivers

Managed HEAP – CLR

Thread of control

<http://msdn.microsoft.com/en-us/library/8bs2ecf4%28v=vs.90%29.aspx>

Passing by reference – like pointers.

A reference is a symbol (or id) to an object

Frames in the stack – objects inj the HEAP.

We can shrink the size of our libraries with Dynamic-linked libraries (DLLs)

GCC – GNU Compiler Collection

<http://gcc.gnu.org/>

<http://gcc.gnu.org/onlinedocs/>

VB verbose syntax

- Project Euler