Using ASP.NET
(for real?)

Announcements
• Requirements docs
  – nice job!
  – let’s talk about email
• Programming assignment out later today or tomorrow, due 10/26
• Office hours: Thurs 2-4PM, Upson 328
  – if no room in 328, I’ll leave a note with alternate location

C# Crash Course
• Namespaces
  – equivalent of packages in Java
  – using foo; == import foo;
  – namespace foo() == package foo()
• Inheritance
  – like C/C++, use : in class definition
    public class foo : bar {}
C# Property Example

- Class definition
  ```csharp
  public class foo {
    private int fooval = 1;
    public int FooProp {
      get { return fooval; }
      set { fooval = value; }
    }
  }
  ```

- Use
  ```csharp
  foo f = new foo();
  int x = f.FooProp;
  f.FooProp = 3;
  ```

C# Property Comments

- note the lack of parentheses in the property definition:
  ```csharp
  public int FooProp {...}
  ```
- in the set method, value is the "magic" variable containing the new property value
- think of value as a local variable you might declared

C# Collections

- Set of interfaces and classes provided by FCL in Systems.Collections namespace
- Most importantly: Dictionary/Hashtable
  ```csharp
  Dictionary foo; ...
  string x = foo["bar"];   
  ```
- This looks like new language syntax, but it's really just shorthand for
  ```csharp
  string x = foo.Item("bar");
  ```
- The [] operator is overloaded

Preparation

- Install samples
  - create new DB instance called NETSDK, basically repeat SQL Server install
  - run config script
  - more details later
- Create new DB user: ASPNET
  - this is user that IIS runs as
- Grant permissions on database tables for ASPNET user
Demo

- Complete working demo - shop.aspx
- Reverse engineer: what are the new challenges here?
  - using data from DB in page
  - repeated dynamic elements
  - conditional display of elements
  - state management
  - etc!

Demo: Anatomy of VS project

- foo.aspx: a page on your site
- foo.aspx.cs: the C# code for the page
- Global.asax.cs: code shared by all pages
- Web.config: instructions for IIS and the C# compiler
- foo.ascx: custom control
- foo.ascx.cs: code for custom control
- foo.{sql,txt}: can add arbitrary files

Demo

- Fobs.sql, create tables, populate, grant permissions
- login.aspx
  - uses CodeBehind page
  - review of simple dynamic page, events
  - discuss page model?
  - query the DB
  - SQL Injection Attack!

Demo

- login2.aspx
  - same web form, different code
  - fix SQL Injection Attack!
  - use SqlParameter, SqlDataAdapter
  - take a look at FCL Documentation for more info on these classes
Demo

- login3.aspx - use a custom control
- loginPagelet.ascx - create a custom control
  - session state
  - The Session object (instance of HttpSessionState)
  - supports Dictionary-like access: Session["User"]
  - uses cookies! (a little reverse engineering)
  - use asp:panel to group content together
  - set Visible property to show/hide controls

Demo

- shop.aspx - putting it all together
- shopping cart?
  - ShoppingCart.cs - implement a C# class representing a shopping cart
  - Global.asax.cs
    - Session_Start fires at start of new session
    - create ShoppingCart and store at part of Session

Demo

- shop.aspx
  - repeating content
    - DataGrid - display in Grid, some customization possible
    - Repeater - arbitrary customization
    - (DataList) - like Repeater, additional formatting possibilities
  - uses DB as one data source, shopping cart as another

Summary

- Accessing the DB
- Code behind pages
- Custom Controls
- SQL Injection
- Repeating Content
- Session State