

CS193N
Final Exam
Aug. 17, 2007

This exam is open book and notes. No computer of any kind may be used.

Note: your answers may be brief. There is no need to fill all the space provided.

Name (printed) _____

I accept the letter and spirit of the honor code— I have not given or received aid on this exam.

Name (signed) _____

		Score	Grader
C#, .NET Basics			
1.	(7)	_____	_____
2.	(7)	_____	_____
3.	(8)	_____	_____
4.	(8)	_____	_____
5.	(4)	_____	_____
Events			
6.	(12)	_____	_____
ADO.NET			
7.	(8)	_____	_____
8.	(12)	_____	_____
9.	(6)	_____	_____
ASP.NET			
10.	(8)	_____	_____
11.	(8)	_____	_____
12.	(12)	_____	_____
Total	(100)	_____	

Part I: C#/.NET Basics (34 points)

1. (7 points) How does the Common Language Runtime make programming in C# different from programming in a traditional language like C? How does it make it different from programming in Java?

C:

- Type safety
- Memory management, garbage collection
- Threading
- Bounds checking
- Language interoperability
- Reflection

Java

- Language interoperability
- No VM
- Multiple inheritance

2. (7 points) In a C# class, you can write a method that uses the "destructor" syntax. When is such a method called? What is the reason for writing one?

Called at garbage collection time. Used to clean up before removal, e.g., to close files or SQL connections.

3. (8 points) What is the difference between a value type and a reference type in C#? In C#, what is "boxing"?

Reference types are always stored on the heap, and have overhead associated with objects

Value types that are local variables are on the stack and store only the data.

Boxing refers to the automatic conversion of value types to objects when needed for assignment, parameter passing, etc.

4. (8 points) In .NET 2.0, Generics were introduced to the C# language. What underlying problem did this solve?

The basic problem is type safety for collection types and parameters. For example, the compiler can detect attempts to add objects of the wrong type to generic collections.

5. (4 points) What will be the output of the following code? In case you've never tried it, Console.WriteLine applied to a bool prints "true" or "false".

```
String a = new String('a', 10);    //creates a new string of 10 a's
String b = new String('a', 10);

Console.WriteLine(a == b);
Console.WriteLine(a.Equals(b));
Console.WriteLine(Object.ReferenceEquals(a,b));
Console.WriteLine(((Object) a)== b);
//Isn't equals fun?
```

```
true
true
false
false
```

Part II: Events (12 points)

6. (12 points) Shown here and on the next page is skeleton code for the following situation: We are defining a class `Widget` that takes two integers in its constructor. The `Widget` class is supposed to let clients take any special actions that they desire when `Widgets` are created, and it will do this by exposing an event called `WidgetWatch` (the declaration of this is something you will be adding).

`Main` is serving as the client. We would like two things to be done automatically for every `Widget` that is created: call the `Reverse` method, and call the `SetMax` method. To be sure that the methods are called for every `Widget`, you will register these methods as handlers for the `WidgetWatch` event. Also, you will fill in code for these methods so that `Reverse` swaps `x` and `y` in the `Widget` that fired the event, and so that `SetMax` imposes a maximum value of 17 on both `x` and `y`.

One thing to be aware of: `Main` should register for the events once, not every time it creates a `Widget`. So you will have to declare the event accordingly. We have indicated in comments where you are to add code. The output for the widgets constructed below is:

```
[4, 3]
[17, 17]
```

```
namespace Events2
{
    class Program
    {
        static void Main(string[] args)
        {
            //Add code here to register the two
            //methods for the event WidgetWatch

            Widget.WidgetWatch += Reverse;
            Widget.WidgetWatch += SetMax;

            Widget w1 = new Widget(3, 4);
            Widget w2 = new Widget(33, 47);
            Console.WriteLine(w1.ToString());
            Console.WriteLine(w2.ToString());
        }

        static void Reverse(object sender, EventArgs e)
        {
            //Add code for this method here

            Widget w = (Widget) sender;
            int temp = w.X
            w.X = w.Y;
            w.Y = temp;
        }

        static void SetMax(object sender, EventArgs e)
        {
            //Add code for this method here

            Widget w = (Widget) sender;
            w.X = w.X > 17 ? 17 : w.X;
            w.Y = w.Y > 17 ? 17 : w.Y;
        }
    }
}
```

```
public class Widget
{
    //Declare the event WidgetWatch here
    public static event EventHandler WidgetWatch;

    private int _x, _y;

    public Widget(int x, int y)
    {
        _x = x;
        _y = y;
        FireEvents();
    }

    private void FireEvents()
    {
        //Add code here that causes any
        //handlers for WidgetWatch to be called
        if (WidgetWatch != null)
            WidgetWatch(this, EventArgs.Empty);
    }

    public int X
    {
        get { return _x; }
        set { _x = value; }
    }

    public int Y
    {
        get { return _y; }
        set { _y = value; }
    }

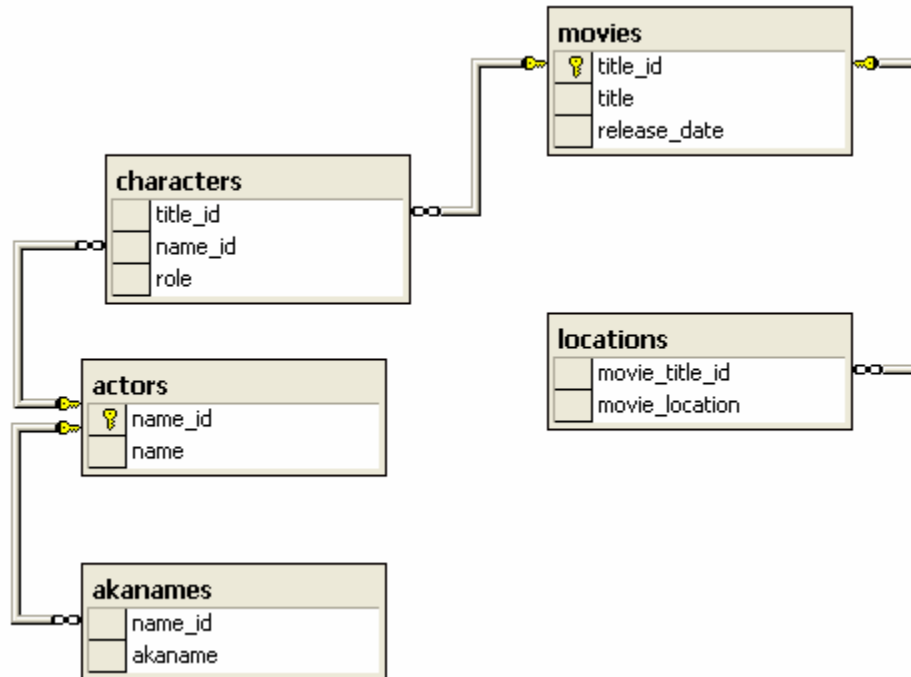
    public override string ToString()
    {
        return "[" + _x + ", " + _y + "]";
    }
}
}
```

Part III: ADO.NET (26 points)

7. (8 points) What is meant by the "disconnected" data model used by .NET? That is, how does the disconnected model basically operate for retrieving data from a database, using the data, and updating the database? What are the strengths and weaknesses of this model?

- An in-memory copy of the data is constructed
- The database must be updated explicitly
- Strength: unload server
- Weakness: doesn't reflect updates from other users, prone to data loss

8. (12 points) For this question you will write some simple SQL queries. The database you will use is similar to (but not quite the same as) parts of the Internet Movie Database (IMDB). Here is a diagram of the database:



Note that the **movies** and **actors** tables have primary keys, and that these are used as foreign keys in other tables. All "id" fields are **ints** and the rest are **strings** except for **release_date** which is an **int** representing the year of release.

In each part below, supply a **SELECT** query per the description. Here is an example:

Retrieve the names of all actors who have played the role 'James Bond'.

```

SELECT    actors.name
FROM      actors INNER JOIN characters
ON        actors.name_id = characters.name_id
WHERE     characters.role = 'James Bond'
  
```

(a) Retrieve a list of all movie names for movies released in the 1980s (i.e., with release date in the range 1980 to 1989 inclusive).

```

SELECT title
FROM movies
WHERE (release_date >= 1980) AND (release_date <= 1989)
  
```

(b) Now, extend the query in part 1 to return only those movies that starred 'Al Pacino' or 'Robert Di Niro'. Return the names of the actors as well.

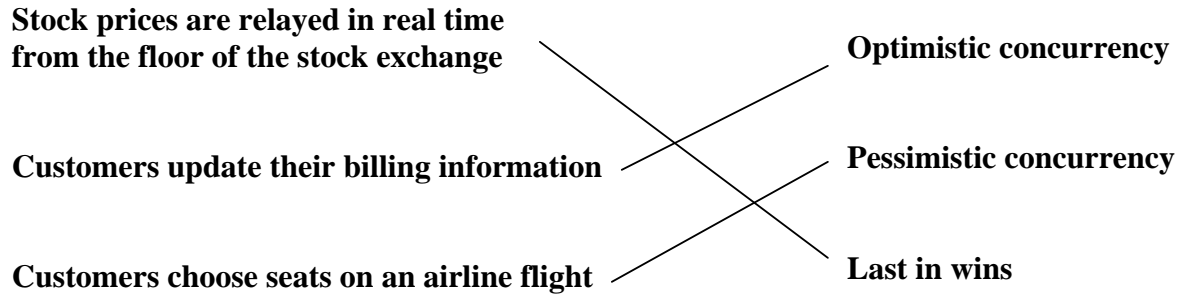
```
SELECT title, name
FROM movies INNER JOIN characters
ON movies.title_id = characters.title_id INNER JOIN actors
ON characters.name_id = actors.name_id
WHERE (release_date >= 1980) AND (release_date <= 1989)
      AND ((actors.name_id = 'Al Pacino') OR (actors.name_id = 'Robert Di Niro'))
```

(c) Retrieve a list of all the locations at which movies containing the word 'Matrix' in their title were shot.

```
SELECT movie_location
FROM movies INNER JOIN locations
ON title_id = movie_title
WHERE title LIKE '%Matrix%'
```

9. (6 points) ADO.NET

On the left below are three situations where a database might need updating, and on the right are three concurrency models. Draw a line from each situation on the left to a model on the right. Given the constraint that no two situations should connect to the same model, choose the best overall match-up. **Briefly justify each line that you drew.**



Justify your lines here:

Stocks: only the latest price matters

Billing: contention not expected for the same customer's records

Seats: high level of contention expected

Part IV: ASP.NET (28 points)

10. (8 points) Suppose a user browses to a ASP.NET webpage, fills in some text boxes, and clicks a button that fires a server-side event. Describe the sequence of actions that takes place on the server.

- build new page
- fill in post back data
- run event code
- send page back to user.

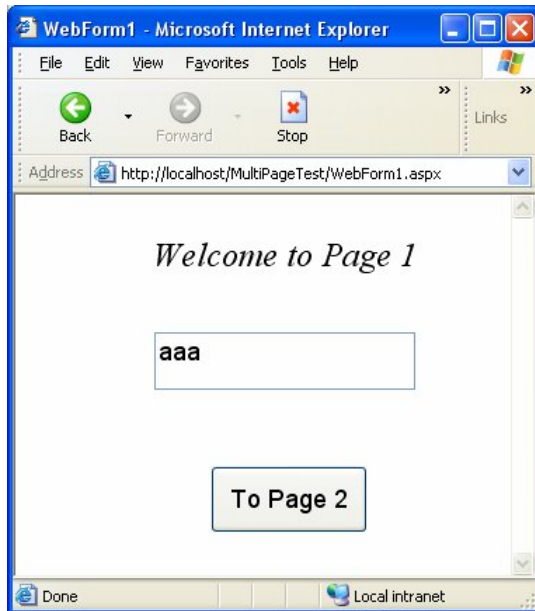
11. (8 points) (a) What is the difference between session state and application state? (b) Can cookies be use to store session state (how, or why not)? (c) Can cookies be used to store application state (how or why not)?

--Session state is information about a particular user; application state applies to all users of the page

--Yes. Server sets values: `Cookies.Response["property"] = <value>;`

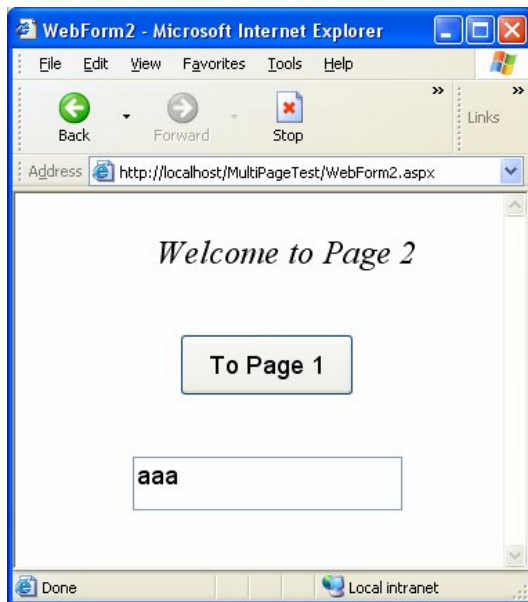
--No. One user's cookie can't reflect another user's activity

Problem 12: (12 points). Consider the pair of web pages shown below. The important code for each page is shown beside it.



```
private void Page_Load(object sender,
                        System.EventArgs e)
{
    if (Session["Text"] != null)
        TextBox1.Text = (string)
                        Session["Text"];
}

private void Button1_Click(object
                           sender, System.EventArgs e)
{
    Session["Text"] = TextBox1.Text;
    Response.Redirect("WebForm2.aspx");
}
```



```
private void Page_Load(object sender,
                        System.EventArgs e)
{
    if (Session["Text"] != null)
        TextBox2.Text = (string)
                        Session["Text"];
}

private void Button2_Click(object
                           sender, System.EventArgs e)
{
    Session["Text"] = TextBox2.Text;
    Response.Redirect("WebForm1.aspx");
}
```

These pages were intended as an example to show the use of session state to move data between pages. The desired behavior is that if the user types "aaa" on Page 1 and clicks the button, the text would appear in the text box on Page 2 (as shown above). If the user then types "bbb" on Page 2 and clicked the button, "bbb" would show up in the text box back on Page 1, and so on.

Your job is to find out whether the code shown above produces the desired behavior.

Part (a) Suppose the user starts up Page 1, types "aaa" in the text box, and clicks the button. What would appear in the text box on Page 2?

“aaa”

Part (b) Suppose the user then types "bbb" in the text box on Page 2 and clicks the button. What would appear in the text box on Page 1?

“aaa”

When a button is pressed, Page_Load runs before the event handler, and in this code, Page_Load does not distinguish the first call from postbacks

Part (c) Suppose the user then types "ccc" in the text box on Page 1 and clicks the button. What would appear in the text box on Page 2?

“aaa”

Same as above