Double-Check Locking for singleton objects

```
public static class Program {
   public static void Main() {
      Singleton s = Singleton.Value;
   }
}

public sealed class Singleton {
  private static Object s_lock = new Object();

  private static Singleton s_value;

  private Singleton() {}

  public static Singleton Value {
      get {
         if (s_value == null) {
              lock (s_lock) {
                if (s_value == null) {
                   s_value = new Singleton();
                }
              }
         }
         return s_value;
      }
   }
}
```

```
public sealed class Singleton
{
     private static Singleton s_value = new Singleton();

     private Singleton() { }

     public static Singleton Value
     {
        get
        {
             return s_value;
        }
     }
}
```

Using the thread pool avoids the overhead of creating dedicated threads.

- All threads in the managed pool are background threads
- There is no way to cancel a work item after it has been queued

Reasons for using a dedicated thread

- The thread needs to run at a special priority
- The thread needs to be a foreground thread
- The thread may need to be aborted

Nullable Types

- Value types can't be null, since they are always allocated as a pattern of bits
- This can cause problems with databases that return Null (meaning missing value)

Nullable Types

```
Nullable <int> i = null;

bool b i.HasValue;  //will be false

int j = i.Value;  //throws an exception

int k = Nullable.GetValueOrDefault<int>(i);  //returns 0
```

Nullable Types in C#

```
int? i = 1;
int? j = 2;
int? k = i + j;
double? d = k;          //implicit cast to double? OK
short? s = (short?) d;  //explicit cast to short?

k = null;
i = k.GetValueOrDefault();   //returns 0
i = k.GetValueOrDefault(2);  //returns 2
i = k ?? 3;                  //returns 3
j = i + k + 4;               //returns null
```

Visual Studio Tools

Profiling (performance testing)

Code analysis

Automated testing

Isdasm (disassembler)

Dotfuscator

Icon editor

Where are we now?

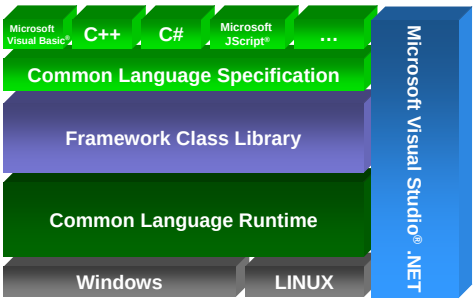
C#
- OO
- Type safe
- Useful syntax
- Events
- Good performance

ADO.NET
- Simplifies DB access
- Strongly typed databases
- Disconnected model
- Works with a variety of providers

ASP.NET
- Visual design of web forms
- Code-behind model
- Server controls
- ASP.NET AJAX

Visual Studio

Microsoft .NET Framework Architecture



Final Exam

Friday, August 17

12:15 – 3:15

Thornton 102