Programming for the WWW

with Emphasis on Java related technologies

Professor Poelman - Fall 2004 NYU
My Philosophy

• Technology churn is such that much of what I teach you today will be obsolete or outdated in a year!
• Architecture changes slower than application technology – understanding architecture is critical
• Java is a must have technology not esoteric
• Java is your next legacy technology
• Teach you to think
Who am I?

• Logan Poelman (pronounced Pole - man)
• You can call me:
  – Logan
  – Professor
  – Professor Poelman
  – Mr. Poelman
Philosophy (cont)

• **You’re all adults** – you’re responsible for what you get out of this class

• **Don’t know?** – ASK!
  – At the appropriate time – think lecture, office hours, email).
  – Utilize your classmates (don’t cheat though).

• You should master, yes, **master** this material!

• **More to life than technology**
  – balance is important
Philosophy (cont)

• **Cheating** - you will get an “F” on that assignment or test. Period.

• **Simple test** - “Would you be embarrassed to tell your parents, friends and ME that’s how you did it?”

• You need to work hard (some harder than others).

• **Hardwork and Frustration** - I am here to work you hard NOT to frustrate you.
My background

• **Project Expertise** – 15+ years experience in industry
  – Acting Chief Architect Chase Home Finance
  – Managing Architect Hotels.com eCommerce Portal
    • BEA Portal
  – Architect on JPMorganChase Global Financial Information cache
    • J2EE + JMS Messaging deployed around the world US, EMEA and Asia
  – Lead Architect on BankBoston Reengineer ($250 MM USD budget)
  – EAI/Messaging Architect Merrill Lynch Direct Markets Web Portal
  – R&D Manager for Electronic Imaging Company
  – Developer of video game Ajax™ on IBM PC 8mHz 8086 1 meg ram,
  – Neuropsychology research in pain – built experimental control / analyzed data
  – Device drivers for DSP60001 board for Mach

• **Director of Consulting** for Valtech Technologies, Inc.
  – [www.valtech.com](http://www.valtech.com) - Office 100 Wall Street

• **Degrees in Computer Science and Psychology**

• **Languages** — Assembly, Basic, DSP Basic, C, C++, Forth, Snobol, Fortran, VisualBasic, Java, C#, VB.net.

• **Platforms** — DOS, Amiga, Windows 3.x/9x/NT/2K/XP, Unix (BSD/Mach/Solaris/Linux), CICS, Java/J2EE

• **DCT (Distributed Computing Technology)**
  – Sockets, RDBMS DB/ODBC
  – COM/DCOM/COM+/.net, CORBA, RMI, EJB/J2EE
  – Messaging (MQ Series, MSMQ, TIBCo, JMS)
Goals of Course

• Understand the architectures of WWW applications
• Understand a selection of technologies
• Understand the value of an OO approach
• Use UML & TDD
• Learn Java and gain experience with Java building web applications
• Learning to think in new ways
Specific Topics & Foci

• Internet Architectures & Protocols
• Full Software Development Lifecycle
  – UP / XP- Inception, Elaboration, Construction and Transition
• Tools
  – NetBeans IDE
  – Java JVM / JRE / JDK
  – JUnit
• Analysis Approaches
  – Object Oriented Analysis
  – Use Cases
• Design Approaches
  – UML – Unfied Modeling Language
  – Design Patterns
  – XML
• Development Approaches
  – TDD - Test Driven Development
  – Java Swing, Servlets, JSPs, WebServices
• Architectures
  – Desktop App
  – Multitier
  – Fat Client - Swing
  – Thin Client - Applets
  – Web – JsP/ Servlets
  – Service oriented Architecture and Web Services
• Real World Application Challenges and Approaches
How many of you have:

• Written Java
• Written C++
• Written C#
• Written HTML
• Written XML
• Written WebServices
• Written UML
• Written Unit Tests
Technology churn –
what was “cool”

• 1997 – HTTP, HTML & Web Servers
• 1998 - CGI & Java
• 1999 - Servlets & EJB
• 2000 – XML
• 2001 – SOAP
• 2002 - WebServices, Struts, Portals & BEEP
• 2003 - WSRP, WSUI, GRID, Pervasive Computing
• 2004 - Java Server Faces, Aspect Oriented, Model Driven Architecture (MDA)
Java is Dead!

• C#, Ruby, Python, … aren’t any better!
• Text based code writing is so 90’s!
• Too complex to deploy a high availability and high scalability system
  – Java + J2EE is more complicated than C++ and CORBA were.
  – Java + J2EE were created and embraced because it simplified C++ and CORBA
• Don’t leave just yet. Still an important language to learn. Not sure what will replace it.
Misc Class Info

• Class Web Page
  – check it every week before class Monday night (Sunday nights after 10PM are a good time to check for updates)

• You needs a CIMS account
  – email csgrad@cs.nyu.edu to request a new account.
Misc Class Info

- **Office Hours**
  - 6-7pm Mondays 4th Floor Warren Weaver Room 401
- **My email**
  - poelman@cs.nyu.edu send me an email 1st week of class from your official email account
  - My MSN IM lp41@nyu.edu
- **Subscribe to class email list**
  - Use the email address you will be sending emails from, else they will likely get bounced.
  - Most notifications will happen through it – pay attention to it
- **Office Phone**
  - 212-998-3081
- **My cell phone**
  - 917-783-7477 (emergencies only)
TAs

• TBD

• TA office hours will be posted by third week of class on the class website.
Class Lecture & Tests

• **All** lecture notes will be available AFTER on the website that lecture (within a day or two).

• **All** homework assignments will be posted by the third week – start the assignments as early as they are assigned.

• Midterm and Final may/may not be open book! I will let you know.

• Check the site after Midnight Sunday, before class EVERY week for updates, changes, notes, etc! – this is **REQUIRED**!
  – Watch the class mailing list – I will send an email after each update to the site
Text Books (See website)

• Required
  – “Learning Java” – O’Reilly
  – Version 2 (J2SE 1.4 NOT J2SE 1.5)
  – Copyright 2002

• Reserved copies at the Courant Library
Suggested Text Books

• “Taming Java Threads” – Holub - Apress
• “Java in a Nutshell” – O’Reilly (basic and J2EE versions)
• “Just Java and Beyond” – Prentice Hall / Sun

• Recommended - Just because they aren’t required doesn’t mean you don’t need to get the info from somewhere – borrow, buy, browse, download and print, xerox a friends copy, hack the code, …
  – Get it with Java 1.4 version info
  – Rule of thumb - Never buy a computer book with a copyright of > 2 years ago, unless no newer version is out
Grading Policy

• 1000 total points
• 5+1 Homework Assignments (5*60 = 300 points total)
  Approximately broken down as follows:
  – 10 points if it compiles
  – 20 points if it runs and matches functionality required
  – 10 point for quality of design
  – 10 points for unit tests and completeness of code
  – 10 quality implementation/structure of code - readability/comments in code
• 2 Exams (300 + 400 = 700 points)
  – Midterm (300 points)
  – Final (400 points) Around 60 questions
  – Multiple choice, possibly matching, possibly short answer
Grading Policy (cont)

• Initial Grading Curve:
  – ≥ 900 → A
  – ≥ 800 → B
  – ≥ 700 → C
  – ≥ 600 → D
  – < 600 → F

• I will curve the grades, if required.
Special Needs

- ESL - English as a Second Language
- Special Needs Students
- Please see me/send me an email, if you need accommodations
Machine Requirements

• Windows 2000 or XP or 2003
• Unix / Linux / Solaris
• At least a 800 mHz processor
• At least 512 megs ram
• At least 1024 x 768 x 16bit color display
• Web and email access
Homework Assignments

• Based on a fictitious mortgage loan company

• **Geeda Home Mortgage (GHM)**

• Lend money to people to buy homes

• I will provide artifacts (documents, diagrams, information) that you will use to understand the problems and build you solutions
Homework Assignments (tentative)

- #0 - Download and install the JDK, JRE and NetBeans IDE (no points)
- #1 - A Swing UI based Java App that is a loan origination application that uses file IO.
- #2 - Now using a multithreaded application layer. Using sockets communicate with “third party system”
- #3 - Swing UI talking RMI to a business logic layer talking to Data Layer - server based 3 tier application.
- #4 - Servlet based UI componentized 3 tier application talking RMI to existing server code.
- $5 – Struts based UI to the existing system. Using WebServices talk to a third party system.
Homework is due:

- Due 14 days after assigned in class (5PM BEFORE that lecture that day)
- See the website for the late policy
- Still need to submit them.
- Submit via the email to your TA.
- Having problems contact your TAs! Or me (try them first).
Homework #0

- You need to download the:
  - NetBeans3.6 + J2SE1.4 bundle
  - Or download the Java 1.4 JDK and JRE + download the NetBeans IDE 3.6
  - You can choose to use another IDE (Eclipse for instance) but you must build the equivalent homework using those IDEs and debug any problems – TAs and I can’t help much, then.

- You need to install them
- You need to create a simple java program called “HelloWorld”
- You need to do this by Sept 20th!
Start NetBeans and choose “New” off the file menu
Click “new” on NetBeans objects

Create a new NetBeans object, for example:

- A Swing Form
- An Applet
- A JavaBean
- A Java Server Page

Or, select from the complete set of object templates.

You can double-click a method or variable in the Explorer to open the Source Editor and jump straight to that point in the source.
Go to “classes” and choose “main”
Then give it the name “HelloWorld” (exact case)
Click “next”
Click “next”
Click “next”
NetBeans generates a class called "HelloWorld.java"

/*
 * HelloWorld.java
 *
 * Created on September 7, 2002, 2:26 PM
 */

/**
 * @author logan.poelman
 */

public class HelloWorld {

/** Creates a new instance of HelloWorld */
 public HelloWorld()
 {

 }

/**
 * @param args the command line arguments
 */
 public static void main(String[] args)
 {
 System.out.println("Hello NYU");
 }


Add this lines to the file generated by NetBeans
Then go to the “Build” menu and select “Execute” (or press F6). It will compile your `HelloWorld.java` file into a file called `HelloWorld.class` and then execute it invoking the method called “`main()`”
Files generated by the NetBeans wizard
And you should compile your java file and then execute it from a command prompt (or shell).
useful URLs

- http://www.cs.nyu.edu/courses/fall04/G22.3033-009
- http://www.cs.nyu.edu/mailman/listinfo/g22_3033_009_fa04
- http://www.netbeans.org/