Programming for the WWW

with Emphasis on Java related technologies

Professor Poelman - Fall 2005 NYU

All Rights Reserved
Goals of Course

• Help you to become better software designers
• Improve your thinking skills –
  – Requirements definition, analysis, abstraction, design, coding, testing, debugging, refining, …
• Teach you to *think* in new ways
• WWW
  – Understand the *architectures* of WWW and *design* of WWW Applications
  – Understand a WWW related technologies
    • IP, TCP, HTTP, XML, WebServices, …
• Design Patterns
  – Command, Adapter, Listener, Singleton, RPC, …
• Understand the *OO (Object Oriented)* approach to software design
  – Abstraction, Encapsulation, Inheritance, Polymorphism, Generics
• Use UML as a diagramming tool
• TDD (Test Driven Development) – to help build more robust code and support refactoring
• Learn *Java* and gain experience with Java building applications
  – I will **NOT** be teaching you Java
    • I’ll teach you concepts, approaches, ideas, technologies, directions
    • You learn by reading, thinking, coding and testing
    • You should try to find the answers to questions on your own before asking me – you’ll learn more that way
  – I will cover important concepts, language, idioms, frameworks, design patterns and ways to devise solutions
    that are implemented using a tool, in this class that tool is java.
• You’ll use NetBeans IDE 4.1 + Java 5.0 (JSK 1.4 update 4)
Specific Topics & Foci

- Internet Architecture(s) & Protocols
- Full Software Development Lifecycle
  - Industry Methodologies - UP / XP - Inception, Elaboration, Construction and Transition
- Java
  - JRE / JDK 5.0 (1.5) + Java JVM
  - The syntax and grammar of Java, JRE, JDK Swing, Sockets, Threads, File IO, Servlets, JSP, JUnit, RMI
- Tools
  - NetBeans IDE 4.1, JUnit, ArgoUML (Maybe)
- Analysis Approaches
  - Object Oriented Analysis, Use Cases, UML
- Design Approaches
  - Design Patterns
- Development Approaches
  - TDD - Test Driven Development
  - Desktop UI versus Web base UI, client/server, peer to peer
- Design Patterns
  - Desktop App
  - 3 tier, n-tier, Multitier
  - UI - Fat Client – Swing, Thin Client – Applets, Web Browser based – JSP/ Servlets
  - Service oriented Architecture and Web Services
- Real World Application Challenges and Approaches
Me?

• Logan Poelman (pronounced Pole - man)
• You can call me:
  – Logan
  – Mr. Poelman
  – Or whatever you like (please be polite ;-)
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 Design – understanding architecture is critical
• Java is a must have technology **not esoteric**
• Java is your **next legacy technology**!
  – just like assembly language, cobol, fortran, basic, powerbuilder, Ada, Tcl, PERL, C, C++ …
  – sooner or later you have to learn a new tool.
• Teach you to **think not just teach you a tool**
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
    • e.g., lecture, office hours, email).
  – Utilize your classmates to clarify questions – not to get answers i.e., don’t cheat

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

• **More to life than technology**
  – balance is important
  – fun outside of school makes you more productive at school
Philosophy (cont)

• **Cheating**
  – you will get an “F” in the course. Period.
  – See the website for details, if you are confused.
  – You don’t need to cheat to succeed in this course
  – **Simple test** - “Would you be embarrassed to tell your parents, friends and ME that’s how you did it?” If so, its probably cheating

• **Hardwork vs. Frustration**
  – You need to **work hard** (some harder than others).
  – You’ll struggle, read and reread and write lots of code. Keep trying
    • the results are worth it as testified to by previous students
  – You will learn more in this course than many seasoned professionals know about Java and WWW Design
  – I am here to work you **hard** NOT to **frustrate** you.
My background

• **Project Expertise** – 15+ years experience in industry  
  – Current - Manager of Infrastructure Architecture and Standards for JP Morgan Chase  
    • Worldwide Security Services, Treasury Services, Commercial Bank and Central Transaction Operations Infrastructure  
    • 3200 Applications in production that I have to worry about  
    • Currently working on a initiative involving 64/128 way SMP machines running AIX 5.3 for Server Consolidation (machine price $12 million each )  
  – Chief Architect Chase Home Finance  
  – Director of Consulting for Valtech Technologies (1200 people worldwide) - NYC  
  – Managing Architect Hotels.com eCommerce Portal - BEA Portal  
  – Managing Architect BureauVeritas Knowledge Portal – IBM Portal – Onshore and offshore team (Team in India, France, Texas, NYC and NJ)  
  – Architect on JPMorganChase Global Financial Information cache - J2EE + JMS Messaging deployed around the world US, EMEA and Asia  
  – EAI/Messaging Architect Merrill Lynch Direct Markets Web Portal  
  – Lead Architect on BankBoston Reengineer ($250 MM USD budget)  
  – R&D Manager for Electronic Imaging Company – Document Management System – 14" Optical Disk Platters, XRay scanners  
  – Developer of video games Ajax™ and RoboWarrior™ on IBM PC (8mHz 8086 1 meg ram!)  
  – Neuropsychology research in pain – built experimental control / analyzed data  
  – Device drivers for DSP60001 board for Mach OS  

• **Degrees in Computer Science and Psychology**  

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

• **Platforms** — Unix (BSD/Mach/Solaris/Linux/AIX), CICS, Java/J2EE, POWER5+AIX, Windows 3.x/9x/NT/2K/XP/2003, MS DOS, Amiga,, AIM65  

• **Computing Technology (experience/knowledge of)**  
  – App Servers – WebSpher, WebLogic, JBoss, Tomcat, Apache  
  – Network Communications – Ethernet, SONET, FDDI, SMDs, Infiniband, VIA  
  – RDBMS – Oracle, Sybase, SQL Server, DB2, IMS, Raima, MS Access  
  – RPC/ORB - Sockets, RMI, CORBA, COM+, .net, CORBA, RMI, J2EE, WebServices  
  – EAI – Messaging, MQ Series, MSMQ, TIBCo, JMS  
  – Processor Architectures – IA32, IA64, POWER, SMP CC-NUMA, CMT,CMP  
  – Storage – SAN, NAS, DAS, iSCSI, Optical Disk JukeBoxes  
  – Protocols – HTTP, FTP, SAMTP/POP3/IMAP, WebServices, WSRP, JSR168, WSUI, BPEL, 802.x, TCP/IP  
  – Markup Languages – HTML, XHTML, XML, XSL,
How many of you have:

• Written Java?
• Written C++?
• Written C#?
• Written PERL?
• Written Python?
• 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, JSP
- 2001 – SOAP, Linux, Message Driven Beans (MDB)
- 2002 - WebServices, Struts, Portals & BEEP
- 2003 - WSRP, WSUI, GRID, Pervasive Computing
- 2004 - Java Server Faces (JSF), Aspect Oriented Programming (AOP), Model Driven Architecture (MDA)
- 2005 – SpringFramework, CMP (Chip Multi Processor), CMT (Chip Multi Threading), iSCSI, Server Virtualization
Java is Dead!

- I love Java – *don’t get me wrong*
- Too complex to deploy a high availability and high scalability system
  - Still lots of people are doing this right now!
  - Java + J2EE were created and embraced because of the work required by C++ and CORBA solutions
  - Web + Java + J2EE are now more complicated than C++ and CORBA were. (that means a change is due)
  - Best option we currently have!
- C#, Ruby, Python, … aren’t any better!
- Text based code writing is so 90’s!
- Things that might be the next wave
  - MDA, Aspect Oriented Prog, CURL, AJAX, Grid, ObjectSpaces, …
- *Don’t leave just yet.*
  - Java is still an important language to learn (the most important).
  - Widely used throughout industry and academia
  - It will be replaces (5-7 years) - Just not sure what will replace it.
Geopolitics, Economics and Philosophy -
According to Me

- Multinational Corporate Capitalism (MCC) has more power and money than Nation States – USA, Japan, Europe even China
- MCC doesn’t care about people (it can’t) – you must care about you and your community
  - PS I am a capitalist (More Adam Smithian than MCC, though.
  - Not a communist, socialist. A bit of an anarchist, though ;-)

- **Programming is dead!**
- Coding is going offshore (and will continue to)
  - 1st India, then Russia, then China, then Bangladesh, then Pakistan, then Afghanistan, then …
    - "Warning the Indian IT industry against complacency, the head of Satyam Computer Services has said it needs to continuously innovate to keep ahead of other developing nations including China which are "eyeing a piece of the pie .. other nations could overtake it", - Satyam Chairman B. Ramalinga Raju - [http://www.deccanherald.com/deccanherald/apr72005/update95526200547.asp](http://www.deccanherald.com/deccanherald/apr72005/update95526200547.asp)
  - Coding is a commodity skill, like oil, it will be bought from the lowest cost producer
  - Cost of coding is the cost of infrastructure + wages – go were cost of living is low and wages are low, until wages start to increase
  - Cost of real estate in Bangalore, India (major offshore development center) is rising fast
    - "Says a senior spokesperson of Bangalore-based Puravankara Projects, "Appreciation in price has been almost 70 per cent in residential apartments, and land values have gone up by over 100 per cent in some cases." Residential prices of new projects have been upped as many as three times in the last 12 months, despite which 60 per cent space is already pre-sold." - [http://us.rediff.com/money/2004/nov/08spec.htm](http://us.rediff.com/money/2004/nov/08spec.htm)

- **Job hopping in Bangalore is about 30%-40% per year in Bangalore.** That means India is starting to cost more to develop in, that means the money will go to other lower cost places sooner or later (sorry India but you are next to lose coding jobs)
  - Automation will replace much coding after that – software will write software – cheaper and it will come back onshore – see technologies like MDA
    - Example – auto industry – went to Mexico and SE Asia, now much is done by robots, here in the US
    - Example fashion – no seamstresses in the US anymore but we still design much of the worlds fashions
Geopolitics, Economics and Philosophy - According to Me

• But he good news is **Design and Architecture are in demand**
• Get a PhD and you aren’t a commodity developer – think about higher abstractions and harder problems
• Get **domain knowledge**
  – financial industry, healthcare, fashion, pharmaceuticals, manufacturing, genetics, nanotech, …
• Help to define the problem and the solution (not just implement it) and you’ll be in demand
• Get psychology, human factors, sociology, public speaking, management and business courses, too.
• Be entrepreneurial and you’ll not be commoditized
  – create new companies, products, markets, technology and solutions

• Why tell you this? To encourage you to do more in CS and not give up!!!!
You need more than just CS

“Interdisciplinary education in computational science and computing technologies is inadequate, reflecting the traditional disciplinary boundaries in higher education.

– Interdisciplinary computational science research and education would also benefit from inclusion of the social sciences and humanities, particularly as complex scientific and engineering problems touch public policy.

– Only systemic change to university organizational structures will yield the needed outcomes.”

National Information Technology Research & Development group of the US Government
Misc Class Info

- **Class Web Page**
  - [http://www.cs.nyu.edu/courses/fall05/G22.3033-008/index.html](http://www.cs.nyu.edu/courses/fall05/G22.3033-008/index.html)
  - check it **every week** before class Monday night
    - Sunday nights after 10PM are a good time to check for updates (I tend to work on things on Sunday nights)

- **Class Mailing List**
  - Subscribe

- **You needs a CIMS account**
  - email [csgrad@cs.nyu.edu](mailto:csgrad@cs.nyu.edu) to request a new account.
Misc Class Info

• Office Hours
  – 9-10pm Mondays after class in lecture hall or up in my office (check lecture hall first) 4th Floor Warren Weaver Room 401
  – 8-9PM Wednesdays, by appointment via phone or skype
  – On extraordinary occasions, other days/times by appointment

• My email (best way to contact me)
  – poelman @ cs.nyu.edu
  – send me an email 1st week of class from your official email account
    • Your Name (how to pronounce it, too)
    • NYU SID

• Subscribe to class email list
  – Use the email address you will be sending emails from, or else they will likely get bounced.
  – Most class notifications will happen through it – pay attention to it

• NYU Office Phone
  – 212-998-3081 (during office hours)

• My cell phone
  – 917-783-7477 (for emergencies or really important stuff only, please)
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** that lecture on the website
  - within a day or two of lecture
  - when possible before the day of the lecture.

- **All** homework assignments will be posted on the website before they are assigned
  - start the assignments as early as possible
  - they build on each other so can’t be skipped
  - Plan you time accordingly

- **Check the site** after 10 PM Sunday, before class **EVERY** week for updates, changes, notes, etc!
  - Watch the class mailing list - I will send a class email after each major update to the website to let you know to look
  - This is a course **REQUIREMENT!**
  - Watch the home page for a list of current/previous updates

- Midterm and Final **closed book!**
Text Books (See website)

• Required
  – “Learning Java” – O’Reilly
  – Third Edition (covers J2SE 1.5 / Java 5.0)
  – Copyright 2005
  – By Patrick Niemeyer, Jonathan Knudsen
    $44.95 US
  – Reserved copy at the Courant Library
Suggested Text Books

• **Recommended Texts**
  – “Taming Java Threads” – Holub - Apress
  – “Swing” – O’Reilly
  – On reserve at the Courant Library
  – Just because they aren’t required doesn’t mean you don’t need to get the info from *somewhere* –
    • borrow, buy, browse, search the web, do tutorials on suns site, download and print, xerox a friends copy, hack the code, …

• **Any texts**
  – *Get it with Java 5.0/1.5 version info if possible*
  – *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) – 30%
  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) – 70%
  - Midterm (300 points) – 30%
  - Final (400 points) Around 60 questions – 40%
  - Multiple choice, possibly matching, possibly short answer
  - Closed Book
  - Covers all material we talk about, is assigned or in the homework
  - Cumulative
  - 2 hours long
Grading Policy (cont)

• Initial Grading:
  – $\geq 900 \rightarrow A$
  – $\geq 800 \rightarrow B$
  – $\geq 700 \rightarrow C$
  – $\geq 600 \rightarrow D$
  – $< 600 \rightarrow F$

• I may 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
- NYU – Moses Center
  - http://www.nyu.edu/csd/
No Incompletes!

• Can’t keep up, drop the course
• Very rarely will exceptions will be granted
• Having trouble – contact me or the TA immediately – don’t wait
• The homework comes at a fast pace. Expect 10-20 hours a week of homework!
• Test are hard!
• If you know some Java, that isn’t enough. You will struggle with some aspect of this course.
• Ask classmates to help you understand things you are struggling with
  – no cheating but studying together is a good thing)
This Course Rocks!

• Integrated approach
• Teaches you to think (forces you)
• Helps you to understand the various technology choices you have with Java, Internet and Web
• Learn analysis, design, testing, coding, refactoring and integration
• Real world based approach – not theoretical
• You will be very sharp on various Java technologies, OOAD, TDD, UML, Internet, Web, etc. when you finish this course
Machine Requirements

- Windows 2000 or XP or 2003
- Unix / Linux / Solaris / AIX / Mac OS X
- If Intel CPU, at least a 1 GHz processor
  - 2 GHz probably
- At least 512 megs ram
  - 1 gig probably best
- At least 1280 x 1024 x 16bit color display
  - Dual monitors is the best
- Web and email access
Homework Assignments

• All are based on a fictitious (fake) mortgage loan company called **Geeda Home Mortgage**
  – “We lend money to people so they can by a house.”

• I will provide artifacts (documents, diagrams, information) that you will use to understand the problems and build your 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
  – Even if late, you need to submit all homework
• Submit via the email to your TA.
• Having problems? contact your TA!
  Or contact me (but try the TA first).
Homework #0

- [http://www.cs.nyu.edu/courses/fall05/G22.3033-008/homework.htm](http://www.cs.nyu.edu/courses/fall05/G22.3033-008/homework.htm)
- [http://www.cs.nyu.edu/courses/fall05/G22.3033-008/homework_assignment_0.htm](http://www.cs.nyu.edu/courses/fall05/G22.3033-008/homework_assignment_0.htm)

**You need to:**
- Download and install J2SE1.5 update 4 (JDK not the JRE)
  - See class website for details
  - Java 5.0 JDK + NetBean 4.1 IDE available at [http://java.sun.com/j2se/1.5.0/download-netbeans.html](http://java.sun.com/j2se/1.5.0/download-netbeans.html)
- You must use Netbeans for this course (sorry Eclipse fans)
- This way the TAs and I can help you with problems better

**You need to install them**

**You need to do several tutorials from the NetBeans IDE**

**You need to do this by Sept 20th!**
Install NetBeans 4.1 IDE (written in Java, BTW)
NetBeans Tutorials

- Then do the **General Java application or library**
  do all the tutorials on that page
  - Setting Up a Project
  - Creating and Editing Java Source Code
  - ...
  - Customizing the Build Process
And then send an email to me with the output

- init:
  - deps-clean:
    - init:
      - deps-clean:
        Deleting directory C:\_NYU\JavaCode\MyLib\build
        Deleting directory C:\_NYU\JavaCode\MyLib\dist
    clean:
    Deleting directory C:\_NYU\JavaCode\MyApp3\build\production
    Deleting directory C:\_NYU\JavaCode\MyApp3\dist
    init:
    - deps-jar:
      - init:
        - deps-jar:
          Created dir: C:\_NYU\JavaCode\MyLib\build\classes
          Compiling 1 source file to C:\_NYU\JavaCode\MyLib\build\classes
          compile:
          Created dir: C:\_NYU\JavaCode\MyLib\dist
          Building jar: C:\_NYU\JavaCode\MyLib\dist\MyLib.jar
          jar:
        Created dir: C:\_NYU\JavaCode\MyApp3\build\production\classes
        Compiling 1 source file to C:\_NYU\JavaCode\MyApp3\build\production\classes
        Compiled 1 empty directory to 1 empty directory under C:\_NYU\JavaCode\MyApp3\build\production\classes
        compile:
        Created dir: C:\_NYU\JavaCode\MyApp3\dist
        Building jar: C:\_NYU\JavaCode\MyApp3\dist\MyApp3.jar
        jar:
        BUILD SUCCESSFUL (total time: 0 seconds)
useful URLs

- http://java.sun.com
- http://www.cs.nyu.edu/mailman/listinfo/g22_3033_008_fa05
- http://www.netbeans.org/