Computing in Finance

Instructor: Lee Maclin
lee.maclin@gmail.com

Instructor: Eran Fishler
eran.fishler@gmail.com

TA: Tingliang (Edward) Guo
tingliang.guo@nyu.edu

Group web site

- Subscribe – Send email to:
  - cif2013-subscribe@yahoogroups.com
- Group home page
  - http://groups.yahoo.com/group/cif2013
- Post message
  - cif2013@yahoogroups.com

Group web site

- Use the group web site to discuss all class and homework related questions
- Do not email the TA and the instructors directly – This doesn't help others to understand the problem
- Do not post solutions to the web site. Ask questions that, if answered, can help you solve the problem.
- Do not ask questions that have to be answered by actually solving the problem for you.

Purpose of this class

- Primary purpose: To teach software development in a financial context
- Secondary purpose: To introduce financial concepts that are best discussed in terms of computing. Examples:
  - High frequency data and trading
  - Back-testing and simulation
  - Order book mechanics
- This is not a class for learning basic Java. You can do that on your own

Books

See *links* section on group site

Why no single text book?

Look at the syllabus!
We draw on too many disciplines.

Can I really learn basic Java in 24 hours?

- You can learn basic Java in 12 hours
- You will spend the next five years improving your programming skills
- In this class, we will cover what the book tells us should be covered in 7 hours
- Best way to learn Java: Two passes through the book
  - First pass: Skim
  - Second pass: Do the examples
- If you are new to computing, do it this weekend or you will fall behind when the classes start

Why Java?

- Fast execution – Very close to C++
  - Worst case scenario: Slower execution than C++ by factor of 2
    - http://scribblethink.org/Computer/javaCbenchmark.html
- Faster development time than C++ by a factor of 5!
  - For most financial applications, development time is more important than running time.
  - For applications where that is not true, you can always optimize with native code
- Easier to learn – Takes less time to make developers productive
- Widely used – Experiment?

Homework

- Due at 7pm on due date unless otherwise specified
- Format of file to send:

  <nameOfAssignment>_<yourId>_<version>.tar.gz
- We won't accept mis-named assignments
- When you send a file to me, to Eran, and to the TA, the subject line should be the above name
- The TA will respond promptly with “RECEIVED” or “MISNAMED”

Homework

- Inside the tar.gz package should be a README.TXT file explaining how to run your code
- All code has to run correctly on Courant's machines
- If it does not, we will take points off
- There should be no binaries or .class files in your tar.gz
- We may provide sample files on which you can test your homework. If so, don't send them back to us
- Homework is not a group exercise. Do your own homework. We will check.

Homework

- If we discuss a homework assignment in class, there can be no late submissions
- If we do not discuss a homework assignment in class, submitting one day late costs you 1/3 of your grade
- Submitting two days late costs you 2/3 of your grade
- In grading your homework, we will drop your worst grade
- There will be no make-up assignments
- If you miss a class, it is your responsibility to use the group site and your friends to figure out what you need to do to submit the homework assignment

Final Grade

- Your final grade will be comprised of the following
- 40% homework
- 40% final exam
- 20% quizzes
- Quizzes will be short and always cover only what was previously discussed in class or assigned as homework
- The worst quiz grade – just like the worst homework grade – will be dropped
- There will be no make up quizzes

Logging in from various platforms

- You can come here and work on the machines in the labs (not recommended)
- You can log in from a Linux / Unix / Mac command line as follows: ssh –l maclin math1.cims.nyu.edu
- Note: Do not use the above machine! Use the machine NYU computing services tells you to use
- You can get a Unix like shell for your Windows machine – Might as well be Git

Git shell for Windows

- Do a Google search for git-1.7.4
  - Git-1.7.4-preview20110204.exe
- Install it
- What you get:
  - perl
  - git
  - ssh
  - scp
  - ..more
- Example of using scp to move files to your Courant account: scp <from> maclin@math1.cims.nyu.edu:<to>

Get info on Courant's machines here

http://cims.nyu.edu/webapps/content/systems/userservices/netaccess/tunnel

Integrated Development Environments

- IntelliJ IDEA
  - http://www.jetbrains.com/idea/
  - What Eran uses. Probably your best choice.
- Eclipse
  - www.eclipse.org
  - What I use (out of habit)
- NetBeans
  - The instructions for installing NetBeans are in the appendix of your Java book
- On your own computer, in addition to an IDE, you will have to install the Java runtime environment (JRE), and the Java Development Kit (JDK)

Starting a project in your IDE

- We will make two source directories
  - src
  - junit
- Add junit to your libraries
- Create a package called ‘lecture1’ in source directory ‘src’
- Create a package called ‘lecture1’ in source directory ‘junit’
- Setting command line arguments for HelloWorld.java