CS 891: Introduction to Parallel Java Programming

Overview & Logistics

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Lesson

• Understand the course topics & logistics
  • Course philosophy
  • Course contents
  • Structure of the lecture material
  • Overview of the assignments & assessments
• Setting up the Java & Android IDE on Android Studio
• Accessing Android & Java source code
Course
Philosophy
There's a growing need for software developers who know how to write parallel programs for a range of computing platforms:
- e.g., mobile devices, laptops, desktops, & cloud environments.
Course Philosophy

- Demand is driven by software/hardware infrastructure advances
  - e.g., multi-core & many core processors, mass storage, ubiquitous network connectivity, & commodity hardware & software platforms

See www.gotw.ca/publications/concurrency-ddj.htm
Course Philosophy

- Effective techniques & practices for developing parallel programs & mobile apps are *not* best learned through generalities & platitudes

“Sitting & thinking” is not sufficient…
Instead, it’s better to see by *example* how these programs can be made
• *easier* to write & read,
• *easier* to maintain & modify,
• *more* efficient & resilient by applying time-proven software patterns & object-oriented & functional design & programming techniques

This course involves a lot of hands-on software development & testing
Summary of the Course Contents
Summary of Course Contents

• Key Java parallelism frameworks

  - Fork-Join Pools

  - Parallel Streams

    - filter(not(this::urlCached))
    - map(this::downloadImage)
    - flatMap(this::applyFilters)
    - collect(toList())

  - Completable Futures

    - /page\ = 8
      - supplyAsync
        - (getStartPage())
      - /imgNum\ = /page\ 8
        - .thenApplyAsync
          - (countImages(page))
        - .thenApply(List::size)
      - /imgNum\ = /page\ 8
        - .thenComposeAsync
          - (crawlHyperLinks(page))

Including Java 8 object-oriented & functional programming language features
Summary of Course Contents

- Key Java parallelism frameworks
- Some Android UI & inter-process communication (IPC) mechanisms

See www.coursera.org/specializations/android-app-development for more instruction on Android
Summary of Course Contents

• Key Java parallelism frameworks
• Some Android UI & inter-process communication (IPC) mechanisms
• Some mobile & web communication mechanisms
Summary of Course Contents

- Key Java parallelism frameworks
- Some Android UI & inter-process communication (IPC) mechanisms
- Some mobile & web communication mechanisms
- Patterns/frameworks for parallel programming
Summary of Course Contents

• Key Java parallelism frameworks
• Some Android UI & inter-process communication (IPC) mechanisms
• Some mobile & web communication mechanisms
• Patterns/frameworks for parallel programming

• We assume you know (or can quickly learn) Java, Android, & Git

See [www.coursera.org/specializations/android-app-development](http://www.coursera.org/specializations/android-app-development)
Structure of the Lecture Material
Structure of the Lecture Material

- This course has four main modules

<table>
<thead>
<tr>
<th>Section</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java object oriented &amp; functional programming features</td>
<td>• Coverage of basic &amp; advanced Java 8 programming features, e.g.</td>
</tr>
<tr>
<td></td>
<td>• Abstraction, inheritance, &amp; polymorphism</td>
</tr>
<tr>
<td></td>
<td>• Lambda expressions, method references, &amp; functional interfaces</td>
</tr>
</tbody>
</table>
### Structure of the Lecture Material

- This course has four main modules

<table>
<thead>
<tr>
<th>Section</th>
<th>Topics</th>
</tr>
</thead>
</table>
| Java object oriented & functional programming features | • Coverage of basic & advanced Java 8 programming features, e.g.  
  • Abstraction, inheritance, & polymorphism  
  • Lambda expressions, method references, & functional interfaces |
| Java Parallelism                       | • Coverage of Java 8 parallelism frameworks, e.g.  
  • Java fork-join framework  
  • Java parallel (& sequential) streams  
  • Java completable futures |
## Structure of the Lecture Material

- This course has four main modules

<table>
<thead>
<tr>
<th>Section</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java object oriented &amp; functional programming</td>
<td>• Coverage of basic &amp; advanced Java 8 programming features, e.g.</td>
</tr>
<tr>
<td></td>
<td>• Abstraction, inheritance, &amp; polymorphism</td>
</tr>
<tr>
<td></td>
<td>• Lambda expressions, method references, &amp; functional interfaces</td>
</tr>
<tr>
<td>Java Parallelism</td>
<td>• Coverage of Java 8 parallelism frameworks, e.g.</td>
</tr>
<tr>
<td></td>
<td>• Java fork-join framework</td>
</tr>
<tr>
<td></td>
<td>• Java parallel (&amp; sequential) streams</td>
</tr>
<tr>
<td></td>
<td>• Java completable futures</td>
</tr>
<tr>
<td>Mobile ⇔ Web Communication</td>
<td>• HTTP communication &amp; parsing libraries</td>
</tr>
</tbody>
</table>
# Structure of the Lecture Material

- This course has four main modules

## Section | Topics
---|---
Java object oriented & functional programming features | - Coverage of basic & advanced Java 8 programming features, e.g.
- Abstraction, inheritance, & polymorphism
- Lambda expressions, method references, & functional interfaces

Java Parallelism | - Coverage of Java 8 parallelism frameworks, e.g.
- Java fork-join framework
- Java parallel (& sequential) streams
- Java completable futures

Mobile ➔ Web Communication | - HTTP communication & parsing libraries

Software Patterns | - Parallel programming & communication patterns
Structure of the Lecture Material

- This course has four main modules

<table>
<thead>
<tr>
<th>Section</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java object oriented &amp; functional</td>
<td>Coverage of basic &amp; advanced Java 8 programming features, e.g.</td>
</tr>
<tr>
<td>programming features</td>
<td>• Abstraction, inheritance, &amp; polymorphism</td>
</tr>
<tr>
<td></td>
<td>• Lambda expressions, method references, &amp; functional interfaces</td>
</tr>
<tr>
<td>Java Parallelism</td>
<td>• Coverage of Java 8 parallelism frameworks, e.g.</td>
</tr>
<tr>
<td></td>
<td>• Java fork-join framework</td>
</tr>
<tr>
<td></td>
<td>• Java parallel (&amp; sequential) streams</td>
</tr>
<tr>
<td></td>
<td>• Java completable futures</td>
</tr>
<tr>
<td>Mobile ⇔ Web Communication</td>
<td>• HTTP communication &amp; parsing libraries</td>
</tr>
<tr>
<td>Software Patterns</td>
<td>• Parallel programming &amp; communication patterns</td>
</tr>
</tbody>
</table>

We will bounce around a bit when covering these topics
Structure of the Lecture Material

• This course has four main modules
  • Each module is composed of lessons
Structure of the Lecture Material

• This course has four main modules
  • Each module is composed of lessons
  • Each lesson is composed of parts
This course has four main modules

- Each module is composed of lessons
- Each lesson is composed of parts
- Each part is a single lecture

Screencasts of each lesson “part” & PDF versions of the slides will be uploaded to www.dre.vanderbilt.edu/~schmidt/cs891f#lectures
Structure of the Lecture Material

• This course has four main modules
  • Each module is composed of lessons
  • Each lesson is composed of parts
• Each part is a single lecture
  • Each part is composed of segments
Structure of the Lecture Material

- There will be a monthly quizzes on material covered in the lectures

Held on the first Wednesday of each month
Structure of the Lecture Material

- There will be a monthly quizzes on material covered in the lectures
- 1st quiz will be next Wednesday

All quizzes (& the final) are “closed book”
Structure of the Lecture Material

- There will be a monthly quizzes on material covered in the lectures
  - 1st quiz will be next Wednesday
  - We’ll hand back & review quizzes at the start of the next class

One of the benefits of a smaller class ;-)}
Structure of the Lecture Material

- There will be a monthly quizzes on material covered in the lectures
  - 1st quiz will be next Wednesday
  - We’ll hand back & review quizzes at the start of the next class

I recommend that you study for quizzes by reviewing slides & watching screencasts available at [www.dre.vanderbilt.edu/~schmidt/cs891f#lectures](http://www.dre.vanderbilt.edu/~schmidt/cs891f#lectures)
Structure of the Lecture Material

- There will be a monthly quizzes on material covered in the lectures
  - 1st quiz will be next Wednesday
- We’ll hand back & review quizzes at the start of the next class
  - If you don’t attend the next class & don’t get your quiz you will be penalized 50%

See [www.dre.vanderbilt.edu/~schmidt/cs891f/work-summary.html#quizzes](http://www.dre.vanderbilt.edu/~schmidt/cs891f/work-summary.html#quizzes)
Structure of the Lecture Material

- There will be a monthly quizzes on material covered in the lectures
  - 1st quiz will be next Wednesday
- We’ll hand back & review quizzes at the start of the next class
  - If you don’t attend the next class & don’t get your quiz you will be penalized 50%
  - Likewise, if you just show up for the quiz & don’t attend class you’ll be penalized 50%

See [www.dre.vanderbilt.edu/~schmidt/cs891f/work-summary.html#quizzes](http://www.dre.vanderbilt.edu/~schmidt/cs891f/work-summary.html#quizzes)
Structure of the Lecture Material

• There will be a cumulative final exam that covers all the lectures

The final exam will be held 2 to 5pm, Monday, December 10th in this room
Overview of the Assignments & Assessments
Overview of Assignments & Assessments

• Programming assignments should be written in Java 8 using Android Studio

You can use any IDE, but your final submission must build/run with Android Studio 3.2.x & Android Pie 9 (API 28)
Overview of Assignments & Assessments

- Programming assignments should be written in Java 8 using Android Studio
- Please install the Java 8 runtime environment (JRE)

See [github.com/douglascraigschmidt/CS891/wiki/Installing-Software](https://github.com/douglascraigschmidt/CS891/wiki/Installing-Software)
Overview of Assignments & Assessments

- Android programming assignments must be submitted using Android Studio
  - A wizard for creating new apps
  - A visual editor for creating GUIs
  - An editor for manipulating Android XML descriptors needed for your app
  - An emulator for testing your apps on your PC
  - A debugger for finding errors in the emulator or on a device

See developer.android.com/sdk
Overview of Assignments & Assessments

- Android programming assignments must be submitted using Android Studio
- Please install Android 9.x Pie (API level 28)

See [en.wikipedia.org/wiki/Android_Pie](en.wikipedia.org/wiki/Android_Pie)
Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub

Go to GitHub at [github.com/douglasraigschmidt/CS891](http://github.com/douglasraigschmidt/CS891)
Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub
- You will need to learn how to use GitLab et al.

GitLab offers git repository management, code reviews, issue tracking, activity feeds and wikis. Enterprises install GitLab on-premise and connect it with LDAP and Active Directory servers for secure authentication and authorization. A single GitLab server can handle more than 25,000 users but it is also possible to create a high availability setup with multiple active servers.

Do you want more from your GitLab installation? A subscription bundles the Enterprise Edition with support from the GitLab team. The Enterprise Edition allows you to sync LDAP groups, control pushes via git hooks, integrate better with Jenkins and Jira, and to run MySQL and forward logs when using our Omnibus package. Our service engineers will help you keep your server running smoothly.
Overview of Assignments & Assessments

- All source code for assignments & exam
  - You will need to learn how to use GitLab et al.
- Be prepared to update your repositories occasionally

“If you don’t like change, you’re going to like irrelevance even less.”
Overview of Assignments & Assessments

- Assignments will provide a range of experience with Java 8 & Android parallel programs

See github.com/douglascraigschmidt/CS891/tree/master/assignments

Keep calm and finish your coursework

See github.com/douglascraigschmidt/CS891/tree/master/assignments
Overview of Assignments & Assessments

- Assignments will provide a range of experience with Java 8 & Android parallel programs
- Implement an image crawler app on Android using various Java 8 features, e.g.
  - Java lambda expressions, method references, & functional interfaces
  - Java fork-join framework
  - Java sequential & parallel streams
  - Java completable futures

The topics covered by the assignments may change a bit during the semester
Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff
Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff
- Assignments *must* be submitted on time or you’ll get a 0

See [github.com/douglascraigschmidt/CS891/wiki/CS-891-FAQ](https://github.com/douglascraigschmidt/CS891/wiki/CS-891-FAQ)
Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff
  - Assignments must be submitted on time or you’ll get a 0
- Your initial submission must compile & be largely complete or you won’t get a review or a final grade
Overview of Assignments & Assessments

• Assignment assessments will be done via reviews by course staff
  • Assignments *must* be submitted on time or you’ll get a 0
  • Your initial submission must compile & be largely complete or you won’t get a review or a final grade
• You will not receive a grade for assignments if you do not attend class regularly

See [www.dre.vanderbilt.edu/~schmidt/cs891s/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs891s/assignments.html)
Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff
  - Assignments *must* be submitted on time or you’ll get a 0
  - Your initial submission must compile & be largely complete or you won’t get a review or a final grade
  - You will not receive a grade for assignments if you do not attend class regularly
- Work *must* be your own
  - This goes for quizzes & programming assignments

www.vanderbilt.edu/student_handbook/the-honor-system#statement-of-the-honor-code
Overview of Assignments & Assessments

- Assessment criteria

<table>
<thead>
<tr>
<th>Assessment Category</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution correctness</td>
<td>40%</td>
</tr>
<tr>
<td>Structure (e.g., modularization, information hiding, etc.)</td>
<td>30%</td>
</tr>
<tr>
<td>Insightful programming (e.g., developing reusable class components, etc.)</td>
<td>10%</td>
</tr>
<tr>
<td>Consistent style (e.g., capitalization, indenting, etc.)</td>
<td>10%</td>
</tr>
<tr>
<td>Appropriate commenting style</td>
<td>10%</td>
</tr>
</tbody>
</table>

See [www.dre.vanderbilt.edu/~schmidt/cs891s/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs891s/assignments.html)
Overview of Assignments & Assessments

• The relative weighting of each portion of the course is:
  • 40% Quizzes
  • 40% Programming projects
  • 10% Final exam
  • 10% Participation

These weightings may change, depending on various factors.
Overview of Assignments & Assessments

• The relative weighting of each portion of the course is:
  • 40% Quizzes
  • 40% Programming projects
  • 10% Final exam
  • 10% Participation
    • Participation is roughly 5% attendance & 5% in-class involvement in discussions
Overview of Assignments & Assessments

• The relative weighting of each portion of the course is:
  • 40% Quizzes
  • 40% Programming projects
  • 10% Final exam
  • 10% Participation
    • Participation is roughly 5% attendance & 5% in-class involvement in discussions

See www.dre.vanderbilt.edu/~schmidt/cs891f/work-summary.html#quizzes & www.dre.vanderbilt.edu/~schmidt/cs891f/assignments.html
Overview of Assignments & Assessments

- The relative weighting of each portion of the course is:
  - 40% Quizzes
  - 40% Programming projects
  - 10% Final exam
  - 10% Participation

- Participation is roughly 5% attendance & 5% in-class involvement in discussions

Don’t expect to get an A in this class if you do not actively participate!!!!
Setting Up the Android & Java IDE on Android Studio
Installing Eclipse Java/Android Developer Tools

- See developer.android.com/sdk

Android Studio
The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

DOWNLOAD ANDROID STUDIO
3.0.1 FOR WINDOWS (683 MB)

- Read the docs  -  See the release notes

- Features  -  Latest  -  Resources  -  Videos  -  Download Options
Installing Eclipse Java/Android Developer Tools

• Installation steps
Installing Eclipse Java/Android Developer Tools

• Installation steps
  • Download & install the Java Standard Edition JDK & JRE 8

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:
- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine
- JDK MD5 Checksum

Looking for JDK 8 on ARM?
JDK 8 for ARM downloads have moved to the JDK 8 for ARM download page.

Java SE Development Kit 8u25

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

- Accept License Agreement
- Decline License Agreement

<table>
<thead>
<tr>
<th>Product / File Description</th>
<th>File Size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux x86</td>
<td>135.24 MB</td>
<td>jdk-8u25-linux-i586.rpm</td>
</tr>
<tr>
<td>Linux x86</td>
<td>154.88 MB</td>
<td>jdk-8u25-linux-i586.tar.gz</td>
</tr>
<tr>
<td>Linux x64</td>
<td>135.6 MB</td>
<td>jdk-8u25-linux-x64.rpm</td>
</tr>
</tbody>
</table>
Installing Eclipse Java/Android Developer Tools

• Installation steps
  • Download & install the Java Standard Edition JDK & JRE 8
  • Download & install Android Studio 3.2.x

[developer.android.com/studio/preview]
Add Components to the SDK

• Launch the Android Studio SDK Manager
• Select “Pie” version of Android (9, API 28)

See developer.android.com/studio/intro/update.html
Add Components to the SDK

• Launch the Android Studio Virtual Device Manager
• Create an Android API 28 emulator

developer.android.com/tools/devices/managing-avds.html
Intel HAXM Virtualization Driver

- Requirements
  - Intel virtualization extensions (VT, VT-x, vmx)
  - AMD virtualization extensions (AMD-v, SVM) [only supported on Linux]
  - Download an x86 emulator image
- Windows & Mac OSX
  - `<sdk>/extras/intel/Hardware_Accelerated_Execution_Manager/IntelHAXM.exe/dmg`
- Linux
  - Install KVM & pass “-enable-kvm” flag to emulator when starting

developer.android.com/tools/devices/emulator.html#acceleration
Accessing Java & Android Source Code
Accessing Java & Android Source Code

• Android source code is available
  • For browsing android.googlesource.com

android Git repositories

To clone one of these repositories, install git, and run:

```bash
  git clone https://android.googlesource.com/name
```

Name
- accessories/manifest
- device/asus/deb
- device/asus/flo
- device/asus/flo-kernel
- device/asus/grouper
- device/asus/tilapia
- device/common
- device/generic/armv7-a
- device/generic/armv7-a-neon
- device/generic/art
- device/generic/common
- device/generic/goldfish
- device/generic/mini-emulator-armv7-a-neon
- device/generic/mini-emulator-mips
- device/generic/mini-emulator-x86
Accessing Java & Android Source Code

- Android source code is available
  - For browsing android.googlesource.com
  - For downloading source.android.com
Accessing Java & Android Source Code

- Java 8 source code is available
Accessing Java & Android Source Code

- Java 8 source code is available
  - For Browsing grepcode.com/file/repository.grepcode.com/java/root/jdk/openjdk/8-b132/java
  - For downloading jdk8.java.net/download.html

---

JDK 8 Project

Building the next generation of the JDK platform

- JDK 8 snapshot builds
  - Download 8u40 early access snapshot builds
  - Source code (instructions)
  - Official Java SE 8 Reference Implementations
  - Early Access Build Test Results (instructions)

Feedback

Please use the Project Feedback forum if you have suggestions for or encounter issues using JDK 8.

If you find bugs in a release, please submit them using the usual Java SE bug reporting channels, not with the Issue tracker accompanying this project. Be sure to include complete version information from the output of the `java -version` command.
Summary
* You will get out of this course what you put into it
Summary

- You will get out of this course what you put into it
- Be prepared to work hard

"Human Felicity is produc'd not so much by great Pieces of good Fortune that seldom happen, as by little Advantages that occur every Day" - Benjamin Franklin
Summary

- You will get out of this course what you put into it
- Be prepared to work hard
- Do *not* miss deadlines...
Summary

- You will get out of this course what you put into it
- Be prepared to work hard
- Do *not* miss deadlines…
- Participate in discussions in class & on piazza

See piazza.com/vanderbilt/fall2018/cs891
Summary

- You will get out of this course what you put into it
- Be prepared to work hard
- Do not miss deadlines...
- Participate in discussions in class & on piazza
- No laptops/phones in class unless explicitly allowed

Failure to comply with this rule will cost you participation points

10 POINTS FROM GRYFFINDOR
• You will get out of this course what you put into it
• Be prepared to work hard
• Do *not* miss deadlines...
• Participate in discussions in class & on piazza
• No laptops/phones in class unless explicitly allowed
• Avail yourself of available resources

See [www.dre.vanderbilt.edu/~schmidt/cs891f](http://www.dre.vanderbilt.edu/~schmidt/cs891f)
Summary

- You will get out of this course what you put into it
- Be prepared to work hard
- Do *not* miss deadlines...
- Participate in discussions in class & on piazza
- No laptops/phones in class unless explicitly allowed
- Avail yourself of available resources

Please resist the urge to email me directly unless it’s a confidential matter or you’d like to schedule a meeting!
Summary

• You will get out of this course what you put into it
• Be prepared to work hard
• Do not miss deadlines…
• Participate in discussions in class & on piazza
• No laptops/phones in class unless explicitly allowed
• Avail yourself of available resources
• There are abundant opportunities!

Summary

• If there’s an emergency, pay attention to the escape route!

• See engineering.vanderbilt.edu/about/evacuationplans.php