CS 253: Parallel Functional Programming with Java, Android, & Spring WebFlux: Overview (Part 3)

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
• Setting up GitLab et al.
• Accessing Android & Java source code
Setting Up the Android & Java IDE on Android Studio
Installing Eclipse Java/Android Developer Tools

- To use Android, you need to install the latest release of Android Studio.

See developer.android.com/studio
Installing Eclipse Java/Android Developer Tools

• Installation steps
Installing Eclipse Java/Android Developer Tools

- Installation steps
- Download & install the latest version of Android Studio

Android Studio Chipmunk | 2021.2.1

On this page ▼
Support for creating your app with Jetpack Compose
Animation Preview supports animatedVisibility
Support for deploying preview from Library Modules
Support for profiling your app
Detect jank on Android 12 and higher
Android Gradle plugin and Android Studio compatibility
Patch releases
- Android Studio Chipmunk | 2021.2.1 Patch 2 (August 2022)
...

Android Studio is the official IDE for Android development, and includes everything you need to build Android apps.

This page lists new features and improvements in the latest version in the stable channel, Android Studio Chipmunk. You can download it here or update to it inside Android Studio by clicking Help > Check for updates (Android Studio > Check for updates on macOS)

To view the release notes for older versions of Android Studio, see Past releases.

For early access to upcoming features and improvements, see the Preview builds of Android Studio.

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

- Launch the Android Studio SDK Manager
- Select Android 12, API level 32

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

• Launch the Android Studio Virtual Device Manager
• Create an Android API 32 emulator

[Image of Android Studio Virtual Device Manager]

[Link to 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

---

**Configure hardware acceleration for the Android Emulator**

- On this page
  - Configure graphics acceleration
  - Requirements
  - Configure graphics acceleration in the AVD Manager
  - Configure graphics acceleration from the command line
  - Enable Skia rendering for Android UI
  - Configuring VM acceleration
    - General requirements
    - Restrictions

The [Android Emulator](https://developer.android.com/studio/run/emulator-acceleration) can use hardware acceleration features to improve performance, sometimes drastically. This page describes how you can configure graphics and virtual machine (VM) acceleration to get higher performance from the emulator.
Setting Up GitLab et al.
Setting Up GitLab et al.

1. Create Your GitLab Repo

www.gitlab.com
git clone git@gitlab.com:your-name/CS-253-fall-2022.git

2. Clone your GitLab repo

Working Folder (Student’s)

3. Change Director into Your Working Folder

cd CS-253-fall-2022

GitLab Repo (Student’s)

See docs.gitlab.com/ee/ssh for info on setting up an SSH key for GitLab et al.
Setting Up GitLab et al.

1. Update from Read-Only GitHub Repo
   git remote add skeletons
git@github.com:douglasraigschmidt/CS253.git
git pull skeletons master

2. Get Current Version
   git pull skeletons master

3. Do work!

4. Commit Changes
   git commit

5. Send Changes to GitLab Repo
   git push origin master

See item #13 at [github.com/douglasraigschmidt/CS253/wiki/CS-253-FAQ](https://github.com/douglasraigschmidt/CS253/wiki/CS-253-FAQ)
Welcome to GitLab
Code, test, and deploy together

Create a project
Projects are where you store your code, access issues, wiki and other features of GitLab.

Explore public projects
There are 1,385,678 public projects on this server. Public projects are an easy way to allow everyone to have read-only access.

Create a group
Groups are the best way to manage projects and members.

Learn more about GitLab
Take a look at the documentation to discover all of GitLab's capabilities.
New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), among other things.

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

To only use CI/CD features for an external repository, choose CI/CD for external repo.

Information about additional Pages templates and how to install them can be found in our Pages getting started guide.

Tip: You can also create a project from the command line. Show command

Project name

CS-253-fall-2022

Project URL

https://gitlab.com/lcfox/

Project slug

CS-253-fall-2022

Want to house several dependent projects under the same namespace? Create a group.

Project description (optional)

Description format

Visibility Level

- Private
  - Project access must be granted explicitly to each user.

- Public
  - The project can be accessed without any authentication.

- Initialize repository with a README
  - Allows you to immediately clone this project’s repository. Skip this if you plan to push up an existing repository.

Create project
Project 'CS-253-fall-2022' was successfully created.

The repository for this project is empty
You can create files directly in GitLab using one of the following options.

- New file
- Add README
- Add CHangelog
- Add CONTRIBUTING

Command line instructions
You can also upload existing files from your computer using the instructions below.

Git global setup
```bash
git config --global user.name "Lindsey Fox"
git config --global user.email "lindsey.fox@vanderbilt.edu"
```

Create a new repository
```bash
git clone https://gitlab.com/lclfox/cs-891-fall-2019.git
cd cs-891-fall-2019
touch README.md
git add README.md
```
Setting Up GitLab et al.

Project members

You can invite a new member to CS-253-fall-2022 or invite another group.

<table>
<thead>
<tr>
<th>Invite member</th>
<th>Invite group</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitLab member or Email address</td>
<td></td>
</tr>
<tr>
<td>Douglas Craig Schmidt</td>
<td></td>
</tr>
<tr>
<td>Choose a role permission</td>
<td></td>
</tr>
<tr>
<td>Maintainer</td>
<td></td>
</tr>
</tbody>
</table>

Read more about role permissions

Access expiration date

Expiration date

Add to project Import

Existing members and groups

Members of CS-253-fall-2022

Lindsey Fox @lcffox it's you

Given access 2 minutes ago

Maintainer
Setting Up GitLab et al.

CS-253 - fall - 2022

Naming, topics, avatar
Update your project name, topics, description and avatar.

Project name
CS-253-fall-2022

Topics
Separate topics with commas.

Project description (optional)

Project avatar
Choose file... No file chosen
The maximum file size allowed is 200KB.

Visibility, project features, permissions
Choose visibility level, enable/disable project features (issues, repository, wiki, snippets) and set permissions.

Project visibility
Private
The project is accessible only by members of the project. Access must be granted explicitly to each user.
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:

```
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
- devicegeneric/armv7-a
- devicegeneric/armv7-a-neon
- devicegeneric/art
- devicegeneric/common
- devicegeneric/goldfish
- devicegeneric/mini-emulator-armv7-a-neon
- devicegeneric/mini-emulator-mips
- devicegeneric/mini-emulator-x86
Accessing Java & Android Source Code

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

The Android Source Code

Android is an open-source software stack created for a wide array of devices with different form factors. The primary purposes of Android are to create an open software platform available for carriers, OEMs, and developers to make their innovative ideas a reality and to introduce a successful, real-world product that improves the mobile experience for users. We also wanted to make sure there was no central point of failure, where one industry player could restrict or control the innovations of any other. The result is a full, production-quality consumer product with source code open for customization and porting.

Governance Philosophy

Android was originated by a group of companies known as the Open Handset Alliance, led by Google. Today, many companies – both original members of the OHA and others – have invested heavily in Android. These companies have allocated significant engineering resources to improve Android and bring Android devices to market.

The companies that have invested in Android have done so on its merits because we believe an open platform is necessary. Android is intentionally and explicitly an open-source – as opposed to a free software – effort; a group of organizations with shared needs has pooled resources to collaborate on a single implementation of a shared product. The Android philosophy is pragmatic, first and foremost. The objective is a shared product that each contributor can tailor and customize.

Uncontrolled customization can, of course, lead to incompatible implementations. To prevent this, the Android Open Source Project also maintains the Android Compatibility Program, which spells out what it means to be “Android compatible” and what is required of device builders to achieve that status. Anyone can (and will!) use the Android source code for any purpose, and we welcome all legitimate uses. However, in order to take part in the shared ecosystem of applications we are building around Android, device builders must participate in the Android Compatibility Program.

The Android Open Source Project is led by Google, who maintains and further develops Android. Although Android consists of multiple subprojects, this is strictly a project management technique. We view and manage Android as a single, holistic software product, not a “distribution”, specification, or collection of replaceable parts. Our intent is that device builders port Android to a device; they don't implement a specification or curate a distribution.
Accessing Java & Android Source Code

- Java source code is available
- For browsing zgrepcode.com

JDK Project

The goal of this long-running Project is to produce a series of open-source reference implementations of the Java SE Platform, as specified by JSRs in the Java Community Process. The Project ships a feature release every six months according to a strict, time-based model, as proposed.

Releases

- 20 (in development)
- 19 (in development)
- 18 (GA 2022/03/22)
- 17 (GA 2021/09/14)
- 16 (GA 2021/03/16)
- 15 (GA 2020/09/15)
- 14 (GA 2020/03/17)
- 13 (GA 2019/09/17)
- 12 (GA 2019/03/19)
- 11 (GA 2018/09/25)
- 10 (GA 2018/03/20)

Resources

- Development list: jdk-dev
- Main-line code repository: https://github.com/openjdk/jdk/
Accessing Java & Android Source Code

- Java source code is available
  - For browsing zgrepcode.com
  - For downloading openjdk.org/projects/jdk

**JDK Project**

The goal of this long-running Project is to produce a series of open-source reference implementations of the Java SE Platform, as specified by JSRs in the Java Community Process. The Project ships a feature release every six months according to a strict, time-based model, as proposed.

**Releases**

- 20 (in development)
- 19 (in development)
- 18 (GA 2022/03/22)
- 17 (GA 2021/09/14)
- 16 (GA 2021/03/16)
- 15 (GA 2020/09/15)
- 14 (GA 2020/03/17)
- 13 (GA 2019/09/17)
- 12 (GA 2019/03/19)
- 11 (GA 2018/09/25)
- 10 (GA 2018/03/20)

**Resources**

- Development list: jdk-dev
- Main-line code repository: https://github.com/openjdk/jdk/
CS 253: Parallel Functional Programming with Java, Android, & Spring WebFlux: Overview (Part 3)