**COS420 – Introduction to Software Engineering – Recommended Tools – Spring 2023**

**Tool Recommendations**

Refer to [Stack Overflow Survey](https://insights.stackoverflow.com/survey/2019#technology) for choosing your platform and frameworks.

**Code Editors and IDEs**

* [Android Studio IDE](https://developer.android.com/studio/)
* [VS Code](https://code.visualstudio.com/) - Versatile and popular code editors with several integrations.
  + GitHub extension - integrates with GitHub repository and allows for easy management of pull requests, branches, etc.
  + Better Comments - highlights comments such as // TODO or // WARN in different colors to allow for better planning and online documentation.
  + Code Spell Checker - identifies misspellings within code and comment
  + General extensions (not language-specific):
    - **Material Icon Theme (by Philipp Kief)** - adds icons for the most commonly used file extensions in most languages; Quality of Life
    - **GitLens (by GitKraken)** - adds Git blame annotations when hovering over a line of code. Misc
  + Language-specific code snippet extensions (exact one depends on the language used) - speed up development by autocompleting common code structures.
    - **ENV (by Jakka Prihatna)** - adds syntax highlighting for .env files.
    - **Markdown checkboxes (by Matt Bierner)** - adds checkbox support for VSCode's built-in markdown preview
    - **YAML (by Red Hat)** - adds YAML language support to VSCode, syntax highlighting, etc.
    - **Bash Beautify (by Ahmed Hamdy)** - adds a formator and syntax support for bash and shell scripts
  + Javascript-specific extensions:
    - **Prettier (by Prettier)** – A formatter for web languages (javascript, typescript, html, css, less, sass, react, vue, etc.)
    - **ESLint (by Microsoft)** - integrates the eslint npm module into VSCode; lints javascript code
  + Other Extensions:
    - **WSL (by Microsoft)** - requires WSL to be installed on the Windows machine; it runs VSCode in the WSL Ubuntu instance instead of Windows. (great for developing in a Unix environment) •
    - **Docker (by Microsoft)** - adds docker syntax support and adds a docker panel to visualize the docker environment
    - **Dev Containers (by Microsoft)** - Docker installs this. It allows you to run VSCode inside of a built docker container, useful for debugging.
    - **Kubernetes (by Microsoft)** - adds a Kubernetes panel to visualize the k8s environment

**Linters and Extensions**

* [Code Climate](https://codeclimate.com/) – Automated code review and test coverage – Place a link on the GitHub site.
* [Code Cov](https://codecov.io/) - Automated code review, quality, and test coverage. Integrates with GitHub repo.
* [SonarQube](https://www.sonarqube.org/) – Automated code quality reviewer – Place a link on the GitHub site.
* [TravisCI](https://travis-ci.org/) – Continuous integration server – Place a link on the GitHub site.
* Similar to TravisCI:
  + GitHub actions <https://github.com/features/actions>
    - GitHub action could run CodeQL to find vulnerabilities on every push/PR to the main branch (it's free for public repos) <https://codeql.github.com/>
  + GitLab Pipelines – free CI/CD tooling (<https://docs.gitlab.com/ee/ci/pipelines/>)
* Cypress <https://www.cypress.io/>
* [Android Lint](https://developer.android.com/studio/write/lint) – Helps identify and correct problems with the structural quality of code.
* [Find Bugs](http://findbugs.sourceforge.net/) – Uses static analysis to find bugs in Java code.
* [PMD](https://pmd.github.io/) – Source code analyzer
* [ZenHub](https://www.zenhub.com/) – Agile Project Management tool for GitHub - Kanban board that integrates with GitHub issue tracking.
* Alternative to Zenhub – Github has projects built-in that allow for the building of sprint logs, Kanban, etc: <https://github.com/features/issues>
* [Flutter](https://flutter.io/) – From Google – For developing Android or iOS apps.
  + Flutter requires to learn Dart, a Google-developed language •
* MySQL Workbench – allows to visually build out/design database tables and gives the SQL table code when the work is done; it's fairly handy for relational database designing.
* DataGrip by JetBrains (<https://www.jetbrains.com/datagrip/>) is also a very useful tool that does everything MySQL Workbench does, but allows students to connect to No SQL databases as well.

**Books**

* [Pragmatic Programmer](https://www.amazon.com/Pragmatic-Programmer-journey-mastery-Anniversary/dp/0135957052/ref=pd_bxgy_14_img_3/134-3602979-2290039?_encoding=UTF8&pd_rd_i=0135957052&pd_rd_r=57910731-cd8c-4b9f-8306-f6e0246baf4d&pd_rd_w=75xUx&pd_rd_wg=Wn6VE&pf_rd_p=09627863-9889-4290-b90a-5e9f86682449&pf_rd_r=STYYG8YY8TEG2Z3NYK8G&psc=1&refRID=STYYG8YY8TEG2Z3NYK8G)
* [Clean Code](https://www.amazon.com/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882?SubscriptionId=AKIAILSHYYTFIVPWUY6Q&tag=duckduckgo-d-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0132350882)
* Head First Android Development - available for free to students through student accounts on
* Many other programming books from O’Reilly – <https://www.oreilly.com>

**Miscellaneous**

* [Android Developer Guide](https://developer.android.com/guide) - A useful resource for general advice on developing Android applications.
* [Docker](https://www.docker.com/) – Container Platform.
* React Tutorial by FreeCodeCamp -
* <https://www.freecodecamp.org/learn/front-end-libraries/react/> - tutorials for other front-end libraries are also available
* Framer.com - free, collaborative, web-hosted UI prototyping
* [Umple](http://cruise.eecs.uottawa.ca/umple/) – Model – Oriented Programming.
* Google Docs and Drive – Students need to invite the instructor to be an editor so that the document revision history may be viewed.
  + - Do not delete or do anything to a document so that the revision history is not wiped out. Rather, move unneeded or older documents to a folder to keep them from cluttering the main directory.
* GitHub – Students need to invite the instructor to be a member.
* OBS Studio – Video recording and live streaming.
* Some tools, such as OneDrive, do not show what is revised and do not keep the complete history.

**Javascript:**

**Recommendation:** Learn first, before NodeJS and React.

**Resources:**  
<https://www.w3schools.com/js/>

<https://www.codecademy.com/learn/introduction-to-javascript> ***(requires a codecademy account)***

**NodeJS:**

**Recommendation:** Learn second or alongside with JavaScript if they are a fast learner.

**Mini description:** Node was developed by Ryan Dahl in an effort to create a JavaScript runtime environment.

**Resources:**

<https://www.w3schools.com/nodejs/>

<https://nodejs.dev/en/learn/>

<https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>

**ReactJS:**

**Recommendation:**

* Learn LAST, learning js/nodejs are both prerequisites to have a slight grasp, React has a steep learning curve.
* Learn NextJS as the React meta-framework – it avoids the hassle of needing to fight with React's native URL router.
* Other React meta-frameworks:
  + RemixJS (<https://remix.run/>)
  + GatsbyJS (<https://www.gatsbyjs.com/>) – this one is very limiting as it is a static site generator; dynamic page rendering is quite difficult;
  + Vite (<https://vitejs.dev/>)
* <https://nextjs.org/docs/getting-started>
* tutorial: <https://nextjs.org/learn/basics/create-nextjs-app>

**Resources:**

* NextJS**:** <https://nextjs.org/docs/getting-started>
* NextJS Tutorial: <https://nextjs.org/learn/basics/create-nextjs-app>
* <https://facebook.github.io/flux/> **<-- React is built on this architecture**
* <https://reactjs.org/docs/getting-started.html>
* <https://reactjs.org/tutorial/tutorial.html>

**Docker:**

**Resources:**

<https://docs.docker.com/get-started/>

<https://medium.com/@kmdkhadeer/docker-get-started-9aa7ee662cea> ***(I skimmed through it, it goes over a fair amount)***

**EXPERIMENTAL TOOLS:**

**Deno:**

**Mini Description:** Deno is a new tool being developed by Ryan Dahl that he introduced at JSConf EU 2018.  His presentation was titled "10 Things 10 Things I Regret About Node.js" and can be found here: <https://www.youtube.com/watch?v=M3BM9TB-8yA>. This tool is to build a most secure runtime environment for JavaScript and also has built-in TypeScript support. I'm currently using it to get ahead of the technology curve as I believe this will grow exponentially just like NodeJS did.

**Resources:**

<https://deno.land/>

<https://www.freecodecamp.org/news/the-deno-handbook/> **<-- Published May 2020, some information in this tutorial is already a little outdated as Deno is constantly updating.**