Set Up Your Environment

- Java
- Eclipse/Spring Tool Suite
- Visual Studio Code
- Tomcat Server
- Git
- Oracle 11g
- SQL Developer

Install Java

- Follow instructions at Oracle to download JDK 1.8.X (as long as it says 1.8, the following doesn't matter. Hence X)
- Double click executable file
- Allow this program to make changes on computer
- Click Next
- *Leave everything default, but notice the installation directory:*
    - C:\Program Files\Java\jdk1.8.X\
- Click Next
- *Will take a bit, be patient.*
- *A new window should pop up.*
- Click Next
- Now you should see:
    - *Java SE Development Kit 8 Update 60 (64-bit) Successfully Installed*
- Click Close

Edit Environment Variables

- Search for "Edit the system environment variables"
- Click "Environment Variables…"
- *You should see two windows*
    - *User variables*
    - *System variables*
- *We care about System variables*
- In bottom window, select "New…"
- Enter values:
    - Variable name JAVA_HOME
    - Variable value C:\Program Files\Java\jdk1.8.X
- Click OK
- In bottom window, find variable with name "Path"

- Click on Path variable
- Click “Edit…”
- Add this line to the end of the Path variable (*be careful here*)
    - %JAVA_HOME%\bin;
- Click OK
- Click OK
- Click OK

Confirm Java Installation

- Open cmd.exe
- Run this command
    - java -version
- *Make sure the output says:*
    - java version “1.8.X”

Install your IDE

- First, what is an IDE? It’s your *Integrated Development Environment* – a software that provides comprehensive facilities to programmers for development. An IDE normally consists of a source code editor with various plugins, build automation tools, and a debugger.
- For Java, we will be using either Eclipse (download plugins later), or a pre-loaded Eclipse called Spring Tool Suites
- No need to download both*
- Regardless of whichever IDE you choose, make sure you’re choosing the enterprise edition and you’re installing it in your C:\Program files folder
- For client side tech, we will be using Visual Studio Code
- Maintaining your IDE…
    - Create a new workspace for each week or related technology
    - A *workspace* is a concept of grouping together a group of related projects and configuration details pertaining to these projects

Install Tomcat Server

- Create “Apache Software Foundation” folder inside of C:\Program Files
- Google Tomcat 8, find apache-tomcat-8.0.36
    - Download zip file
    - Unzip file
    - Move apache-tomcat-8.0.36 folder to C:\Program Files\Apache Software Foundation
- Give full permissions
- Right click – properties
- Security tab
- Edit permissions
- Full control to user

- Apply -> OK
- Create System Variables
  - Catalina_Home
  - C:\Program Files\Apache Software Foundation\apache-tomcat-8.0.36
- To Start Tomcat Independently
  - Navigate to tomcat bin folder
  - Execute startup.bat (Windows batch file)
- To start with Eclipse/STS
  - Select Server View/tab
  - Add new server
  - Select Apache -> Tomcat v8.0 Server -> next
  - Tomcat Installation directory: - navigate to C:\Program Files\Apache Software Foundation\apache-tomcat-8.0.36
  - Next
  - Add any project(s) available (if any)
  - Finish

Setting up our database (Oracle 11g XE)

Note: 64 bit machine is required for this install. Not applicable for MacOS users

- Download at http://www.oracle.com/technetwork/database/database-technologies/express-edition/overview/index.html
- Unzip
- Double click on setup.exe
- Allow program to make changes on computer
- Click Next
- Accept terms
- Click Next
- *Leave everything default, but notice the destination folder is C:\oraclexe\*
- Click Next
- *Leave the ports to their default, but we should write these down:*
  - TNS Port : 1521
  - MTS Port : 2030
  - HTTP Port : 8080
- Click Next
- *Now we will create a password for users SYS and SYSTEM*
- Set your password to : admin
- Click Next
- Click Install
- *This will take a bit so be patient.*
- Click Finish

Confirm installation

- To make sure our Oracle 11g XE Database is running correctly, go to http://localhost:8080/apex
- What you are looking at is called **Oracle Application Express (Apex)** which is a tool for managing an Oracle database. However, there is a competing tool called **Oracle SQL Developer.** We will be using SQL Developer during training.
- Now it is time to set up **SQL Developer.**

Setting up SQL Developer – our Database IDE

- Download at http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/sqldev-downloads-v413-3211385.html
- *Now you should see a folder called* **sqldeveloper-4.1.3.20.78-no-jre**
- Put the **sqldeveloper-4.1.3.20.78-no-jre** directory into **C:\Program Files**
- Go to C:\Program Files\sqldeveloper-4.1.3.20.78-no-jre\sqldeveloper
- Double click on sqldeveloper.exe
- When asked for **jdk location**, enter: C:\Program Files\Java\jdk1.8.0_60
- *SQL Developer should open. I recommend pinning this program to the taskbar.*

Create **Connection** in SQL Developer to SYSTEM user in Oracle 11g XE Database

- Click the green plus sign in the top left of the “Connections” tab
- Set the following values :

Connection Name	SYSTEM – admin
Username	SYSTEM
Password	admin
Hostname	localhost
Port	1521
SID	xe

- Click Test
- *Make sure it says “Status : Success” in the bottom left*
- Click Connect
- *Now you should see a connection called “System – admin”*
- *NOTE: This user should ONLY be used for admin purposes. Do not create tables, sequences, triggers, etc. under this user.*

Setting up the Chinook Database

- *The Chinook Database is a sample database. You will be using this database to initially learn Structured Query Language (SQL). It will be sent in a text file separately*
- In SQL Developer, open SYSTEM’s worksheet.
    - *This is where you may execute SQL commands*

- Copy the contents of **Chinook_Oracle.sql** and paste them into the sql worksheet.
    - *This document contains many sql commands to setup the Chinook database.*
- In the sql worksheet, press Ctrl-a
- Press Ctrl-Enter
- *This will execute the sql commands, and there are a lot of them so be patient.*
- *NOTE: the sql you just ran made a new database user*
    - *Username* chinook
    - *Password* p4ssw0rd
- In SQL Developer, create a new connection to our **chinook** user.

Git Setting up Git

Note: 64 bit machine is required for this install.

- Download Git
- Double click Git-2.7.4-64-bit.exe
- Allow program to make changes on computer
- Click Next
- Click Next
- *Leave default: "Use Git from Git Bash only"*
- Click Next
- *Leave default: "Checkout Windows-style…."*
- Click Next
- *Leave default: "Use MinTTY…."*
- Click Next
- *Leave "Configuring extra options" defaults*
- Click Next
- *This will take a minute.*
- *Note: the destination directory is C:\Program Files\Git*
- Click Finish
- Go to C:\Program Files\Git
- Double click git-bash.exe
- *Git Bash should open. I recommend pinning this program to the taskbar.*
- Type following command (notice the two dashes):
    - git –version
    - Ensure the output looks like: "`git version 2.7.4.windows.1`"

Get GitHub

- Go to http://github.com/
- Set up an account
    - Keep everything professional

Clone Repository

- Open Git Bash
- Type following commands (descriptions to right, bold are git commands)

cd ~	**go to home dir
mkdir my_git_repos	**make new dir (for all repos)
cd my_git_repos	
git clone [batch repository link- will be sent via slack]	****clone repo at <url>**
cd <created directory>	**go into created repo/dir
git checkout -b Firstname_Lastname	****create new branch**
mkdir Firstname_Lastname_Code	**make new dir (for your code)
cd Firstname_Lastname_Code	
touch README.md	**create README.md
echo “Firstname Lastname’s code” > README.md	**put text in file
git add README.md	****add file to staging area**
git commit -m “Firstname Lastname add README.md”	****commit changes**
git push --set-upstream origin Firstname_Lastname	****push changes to central repo**

Helpful Git Commands (for later use)

1. **You made changes to files A.java and B.java and you want to push up the changes**

- git status
    - **you can see the files were modified
- git add A.java
- git add B.java
- git status
    - **you can see the files were added to the **staging area**
- git commit -m “add A.java and B.java”
- git status
    - **nothing to commit, that means the file changes were committed
    - **however, it says “Your branch is ahead of ‘origin/Firstname_Lastname’ by 1 commit.”
- git push
- git status
    - **perfect, now it says “Your branch is up-to-date with ‘origin/Firstname_Lastname’.”

2. **The trainer put new code up on git and you want to pull those changes**
    a. You can look at the code on gitlab.com
    b. This is assuming you want the code locally, the new code is on the master branch, and you are on your Firstname_Lastname branch

- git checkout master
- git pull
- git checkout Firstname_Lastname

- git merge master –m “get new code from master branch”