EECS 2311

Software Development Project

Click to edit Master text styles
Second level
Third level
Fifth level

January 12, 2022

YORK UNIVERSITÉ UNIVERSITY

Overview

- We will develop a large piece of software
- We will learn to use a number of different tools that are designed to help the software development process
- Development will be in groups
  - Everybody is expected to code

Workload

- This course requires 8-10 hours per week per student
- As a group: Make sure to meet every week to plan tasks for each member, and integrate code
- Individually: Study posted material before the lecture (flipped classroom). A lab task will be posted every Wed. You will be able to get help on it from the TAs in the lab and from the instructor during lecture. There will also be 4 individual peer-assessment assignments.

Evaluation (individual)

- 20% - 4 peer assessments
  - For each peer assessment, you will receive either a document or a prototype from another group to provide feedback on
  - Your grade in the peer assessment will be based on the quality of the feedback you provide
- Up to 5% bonus marks for ***accepted*** bug reports on the starter code
  - First-come, first-serve
  - 1% for each high severity bug
  - .5% for each low severity bug

Evaluation (group-based)

- 30% - Midterm submission (due Feb 27)
- 50% - Final submission (due April 11)
- Each group submission will receive a grade based on its merit. Individual grades may be less if full participation has not been demonstrated.

Our project

- An initial description is posted under Project on the course wiki page
- Let's take a look at it…

Intentionally vague requirements

- In a real software development project, requirements are vague and ever-changing
- The exact requirements will be refined iteratively by interacting with the “customer” on an **ongoing basis**
- Some requirements may change after the project has started
- New requirements may be added after the project has started

Groups

- Groups are assigned randomly by the “manager”
- As enrollment in the course changes in the first few weeks, the “manager” may rearrange the groups
- Same as a real software project!

All group members

- Go to github.com and sign up for an account
- If you already have a github account, you can use it for the course
- However, if your existing github account has a username that has nothing to do with your name, you might want to create another account for the course
- **Your participation in the project will be assessed based on your github activity!**

Class “exercise”: Find your group

- Login to the EECS 2311 page on eClass (use your Passport York credentials)
- Find your group number (click on Participants)

Class “exercise”: Find your group

- Go to the breakout room with the same number to find your teammates
- Introduce yourselves and exchange contact information
  - Icebreaker: 2 truths and a lie
- Decide who will create the github repository for the team
  - Gather github usernames if possible

All group members

- On github, click on Settings (in your profile), then Developer Settings → Personal Access Tokens → Generate New Token
- Select for repo access
- Copy the generated token
- When Eclipse asks for your github password, you must instead use this generated token instead

All group members

- Open Eclipse
- Go to Preferences → Version Control (Team) → Git → Configuration
- Click Add Entry, add the pair [**`user.name`**, your name]
- Click Add Entry, add the pair [**`user.email`**, your email]
- These **must** be the same as the ones used at github.com
- Click Apply, then OK

Once per group

- Sign in to one group member's github account
- Navigate to https://github.com/biltzerpos/TAB2XML
- Click on Fork
- Copy the URL of the forked repository
- Click on Settings → Manage Access and add the remaining group members as collaborators
- Your group repository is ready

All group members

- Go to File → Import → Git → Projects from Git
- Click Next, select Clone URI, click Next
- Copy the URL of your repository from github.com on the URI field, and click Next
- Keep clicking next, and finally Finish
- You now have a copy of the project in your local repository

Gradle

- Gradle is a modern build automation tool that a software project can use to automate tasks related to deployment
- Gradle comes installed with the latest version of Eclipse
  - If you don’t have it, you can install it through the Eclipse Marketplace (search for Buildship)

Gradle in Eclipse

- If you cannot see the Gradle Tasks window: Window → Show View → Other → Gradle → Gradle Tasks
- To fix possible compilation errors:
  - Make sure JRE 17 is under Preferences → Java → Installed JREs
  - Preferences → Gradle
    - Select Specific Gradle Version 7.3.3 (or 7.1.1)
    - Under Advanced Options, point Java Home to JDK 17

Gradle in Eclipse

- After making changes to the Gradle Setup, always follow with: Rightclick on the project → Gradle → Refresh Gradle Nature
- In the Eclipse Package Explorer, click on the three vertical dots, select Filters, and uncheck the Gradle build folder, and Gradle sub projects so that they are visible

Gradle code

- Gradle uses a language called Groovy to express the necessary tasks
- We provide a sample **`build.gradle`** file that you can use as a starting point for your project
- For most builds for this course, this is all you will need

Building with Gradle in Eclipse

- In the Gradle Tasks window, expand your project, expand the **`build`** task group, and double-click on **`build`**
- This runs several tasks, such as
  - Resolving dependencies
  - Compiling all code
  - Running your tests

Running with Gradle in Eclipse

- In the Gradle Tasks window, expand your project, expand the **`application`** task group, and double-click on **`run`**
- Results are shown in the Gradle Executions window
- If something goes wrong, detailed information can be found in the Console window

Push

- Make some changes to any of the classes in the project
- Rightclick on any element that has changes (could be the whole project), and select Team → Commit
- Add a commit message
- If you do not want to publish the changes yet, click Commit
- If they are ready to be published, click Commit and Push

Pull

- To get changes published by other group members, right-click on the project, and select Team → Pull
- **Before starting to make changes to the code, it is important to Pull, so that you are working on the latest version**

Until next week…

- Play around with the code and make sure you are comfortable running with Gradle and pushing and pulling from Github
- Study the lab task material that will be posted right after the lecture
- Come prepared with questions to the next lecture. It will be flipped-classroom style, so you will get a lot more out of it if you have looked at the material beforehand.