

Branch: complete-docum... ▾

CS153-Assignments / HW06 / docs / Architecture Overview.md

Find file

Copy path

JeffPersons Docs: Simplified naming d0e59a1 2 minutes ago

1 contributor

39 lines (26 sloc) 1.32 KB

# Simpl ©: Implementation Overview

## Project Structure

```
docs
src
  exceptions
  gen
  main
  utils
  SimplCompiler.java
  SimplMain.java

Simpl      # project directory
  \lib      # third party library jars

  \examples # Simpl programs examples

  \docs     # language overview, diagrams, etc

  \src      # source directory
    \exceptions # collection of exceptions specific to Simpl
    \gen        # antlr generated listener/visitor/parser/tokens files
    \main       # core code including message handling, visitor, jasmin code emitter
    \utils      # various utilities such as files

  \tests     # all Simpl test files go here
    \basic    # small, basic tests
    \comprehensive # larger, more comprehensive tests

simplc.sh   # script to compile a source input (.simpl) file

simplr.sh   # script to run a compiled source (.class) file

build.sh    # script to generate antlr sources, compile codebase, and run tests

...         # readme/simpl scripts/etc files go here in outermost directory
```

Branch: complete-docum... ▾

CS153-Assignments / HW06 / docs / Contribution Guide.md

Find file

Copy path

 JeffPersons Docs: Moved contribution guide to docs 3ae9e51 32 seconds ago

1 contributor

39 lines (33 sloc) 2.38 KB

# Final Assignment for SJSU's CS 153 - Compilers



*Starring Jeff, Itaru, Ben, and Ki Sung*

## WHAT YOU SHOULD ALWAYS DO



- Update from the parent branch AND update current branch before you push (or even commit) anything. **This means git pull and git merge master all the time, before you make/push changes!!**
- *Incremental, staged pull requests AND commits -- that is, never just blindly select all and commit. Rather, if you have a lot of changes, do it in blocks (eg updated readme, improved parser field names, etc).*

## How to contribute using Git from terminal



0. use git clone on the repo, using credentials and this url
1. use `git checkout -b` with `your-new-feature-branch-name` from master to create branch
2. use git add to index the files, commit to update locally, and push to push the changes
3. then do a compare and pull request from the *GitHub website*
4. add the relevant code reviewers, and implement any of their requested changes
5. merge your branch with master - rebase and squash if necessary (test merging on a temporary branch off master if you're not comfortable with that sort of things).

## How to contribute using GitHub Desktop



0. clone the repo into a location into the default GitHub documents folder
1. create Branch
  - i. go `toolbar>branch>new-branch` and then enter *your-new-feature-branch-name*
  - ii. click publish branch (upper middle of the screen)
2. every time you make changes to any of the contents in the *CS153Projects* directory:
  - i. fetch origin (this syncs the remote repositories changes with your local copy)
  - ii. uncheck all changes, and check only the logically related ones to add. IMPORTANT: DON'T LAZILY COMMIT ALL and say LATEST CHANGES, etc
  - iii. add a summary and press commit
3. when you've gotten to the point that you're ready to add your changes to master:
  - i. test your changes, and make sure it compiles with the makefile (you should be doing this incrementally and often, anyways).
  - ii. go onto the GitHub website, go to your branch off of master, and then click compare and pull request
  - iii. give it a title of the change, and summary of why/how/what, and manual/compilation tests you've made, and add reviewers
  - iv. add reviewer to the PR (add everyone if unsure) - and then implement any of their changes

v. once approved, merge with master and delete the feature branch (or someone else will do it for you)