CS 1XA3 Project 01

Due Date: Part 1 Feb 15, Part 2 Feb 26 2019

1 Project Setup

1.1 Create a new folder in your repo

- On the **master branch**, create a new folder **CS1XA3/Project01**

1.2 Add the folowing files

- Add **CS1XA3/Project01/project_analyze.sh**
- Add **CS1XA3/Project01/README.md**

1.3 Commit and Push

- Add and commit with the following message **EXACTLY** "Initial Project01 Commit"
- Push to GitHub

1.4 Create a Branch

- Create a branch called **project01**
- Push the branch to github (i.e **git push origin project01**)
- Work from the **project01** branch and only merge with master when your ready to submit Part 1 or Part 2

2 Part 1 Submission (Due Feb 15)

Merge to master branch a **WORKING SCRIPT** that implements at least **10 points** worth of features

3 Part 2 Submission (Due Feb 26)

Merge to master branch a **WORKING SCRIPT** that implements at least **15 points** (including points from Part 1) and at least one custom feature (describe the feature in the projects **README.md** file)

4 Grading Scheme

- README documentation and ability to follow instructions **15%**
- Custom Feature **25%**
- Other Features **60%** (It is possible to receive up to a **15%** bonus mark for implementing one of the 10 point features and an extra feature)

5 Features

Document each of the features you select and how to use your script in the project **README.md** file

5.1 Script Input (Mandatory) (5 Points)

- Make the script interactive (i.e select which feature is executed) either by system input or by user prompted input (make sure to describe this in your projects **README.md**)

5.2 Create a TODO Log (5 Points)

- Puts each line of every file in your repo with the tag #TODO into a file **todo.log**

5.3 Compile Error Log (10 points)

- Find Haskell and Python files in your repo that fail to compile (i.e have syntax errors) and put them in a file **compile_fail.log** (DO NOT create new files while doing this)
- **HINT**: just compile the files and check for error output

5.4 Merge Log (5 points)

- Find all the commit hashes where merge is mentioned in the commit message and put them in a file **merge.log**
- **HINT**: use **git log –oneline** and the command **cut -d' ' -f1 inp** to get the first column of **inp**

5.5 File Type Count (5 points)

- For each HTML, Javascript, CSS, Python, Haskell and Bash Script file: output a file count for how many of each exist
- i.e example output would be something like HTML: 4, Javascript: 1, CSS: 0, . . .

5.6 Delete Temporary Files (5 points)

- Find and delete all **UNTRACKED** (and untracked only) files ending in the extension **.tmp**
- HINT: **git ls-files . –exclude-standard –others** will list untracked files from the current directory

5.7 Find Last Working File (10 Points)

- For a file given by user input, find the last working version (i.e no syntax errors) and checkout and merge that file to the current branch. If no working version exists, keep the most current version