UF UNIVERSITY of FLORIDA
Herbert Wertheim College of Engineering

COP3504C

Lab 05: Software Engineering

Overview

The objective of this lab is to understand the purpose and usefulness of debugging and version control. Debugging is an essential skill needed to be a successful developer. We will use PyCharm's built-in debugger. We will also practice functions of version control systems to develop familiarity in them.

Specification

You will be debugging your previous solution code in this lab. After you have demonstrated to a lab instructor that you can successfully debug a program, you push a program up to your GitHub repository.

Debugging

1) Open your IDE (PyCharm by default)
2) Open your previous solution (or another lab/project if you didn't complete the previous lab)
3) Demonstrate breakpoints, the watch window, and the stack trace window to your lab instructor (online students may create a short screencast instead)

Git Practice / Setup

1) Complete these exercises: https://try.github.io/levels/1/challenges/1
2) Create a GitHub educational account: https://education.github.com/pack/join
3) Install Git for your OS if necessary. (You can download Git here: https://git-scm.com/downloads)

Optionally, you may also want to install a GUI client for Git and a windowed merge tool. You can get SourceTree (https://www.sourcetreeapp.com/) and Meld (https://meldmerge.org/) for free. (Directions for using Meld with SourceTree: https://jaehoo.wordpress.com/2018/05/21/source-tree-resolve-conflicts-with-an-external-tool/)

Git Tasks

1) Create an empty repository on GitHub and add a README.md (markdown) file. ***Each student should clone***.
2) Individually convert your Java solution from the previous lab (**`analyzer.py`**) into C++ (**`analyzer.cpp`**).
3) Add and commit your converted solution to the GitHub repository. Each student should attempt to push.
4) The second push should create a ***merge conflict***. You should work together to resolve this, then push.
5) Both students should pull the changes, and then the student who does not own the repository should fork it.
6) After forking, each student should set the repository to **private** and add their lab instructor to the repo.

When you convert to C++, you can use the **`std::chrono::system_clock::now()`** function to measure times.

If you are unfamiliar with the command line and would like to learn it, this guide will help (for Mac, Linux, and Git-Bash on Windows): http://www.ee.surrey.ac.uk/Teaching/Unix/

Submissions

For campus students, the debugging section of this lab requires a "check-off" in lab. For UF Online students, this may be submitted instead as a screencast on Canvas. Your GitHub repository can be demonstrated in person (for check-off) or can be checked later.