HOMEWORK WEEK 4

(handout for students)

TASK 1 (Git and GitHub)

Question 1

Complete definitions for key Git & GitHub terminology

GIT WORKFLOW FUNDAMENTALS:

Working Directory

ANSWER:

- The working directory (workspace) is essentially your project folder.
- Also note the term directory is basically synonymous to the term folder.

Staging Area

ANSWER:

- The staging area is a tree file called index that stores information (file, directory, message, ...) about the next commit.
- The index is like a pre-commit.
- The index holds a snapshot of the content of the working tree, and it is this snapshot that is taken as the contents of the next commit.
- Thus after making any changes to the working directory, and before running the commit command, you must use the add command to add any new or modified files to the index (ie staging area).

Local Repo (head)

ANSWER:

- When working with Git, only *one* branch can be checked out at a time - and this is what's called the "HEAD" branch. Often, this is also referred to as the "active" or "current" branch.
- Git makes note of this current branch in a file located inside the Git repository, in .git/HEAD. (This is an internal file, so it should not be manually manipulated!)
- If you wonder what exactly this file contains, you can simply have its contents printed on the command line:

```
$ cat .git/HEAD
ref: refs/heads/master
```

-In this example case, a local branch named "master" is the current HEAD.

Remote Repo (master)

ANSWER:

- In Git, "master" is a naming convention for a branch. After cloning (downloading) a project from a remote server, the resulting local repository has a single local branch: the so-called "master" branch. This means that "master" can be seen as a repository's "default" branch.

WORKING DIRECTORY STATES:

Staged

ANSWER:

- Staged changes are a lot like unstaged changes, except that they've been marked to be committed the next time you run git commit. Upon your next commit, your staged changes become part of your Git history. git status will no longer list them as changes since they're part of your last commit now.

Modified

ANSWER:

- amount of files changed (at least one line added or removed) per commit in a given time-frame.

Committed

ANSWER:

- is an individual change to a file (or set of files). When you make a commit to save your work, Git creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of the specific changes committed along with who made them and when. Commits usually contain a commit message which is a brief description of what changes were made.

GIT COMMANDS:

Git add

ANSWER:

- The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you run git commit. In conjunction with these commands, you'll also need git status to view the state of the working directory and the staging area.

Git commit

ANSWER:

- Commits are the core building block units of a Git project timeline. Commits can be thought of as snapshots or milestones along the timeline of a Git project. Commits are created with the git commit command to capture the state of a project at that point in time. Git Snapshots are always committed to the local repository.

Git push

ANSWER:

- The git push command is used to upload local repository content to a remote repository. Pushing is how you transfer commits from your local repository to a remote repo.

Git fetch

ANSWER:

- In git fetch, it is a primary command used to download contents from a remote repository. git fetch is used in conjunction with git remote, git branch, git checkout, and git reset to update a local repository to the state of a remote. The git fetch command is a critical piece of collaborative git work flows. git fetch has similar behavior to git pull, however, git fetch can be considered a safer, nondestructive version.

Git pull

ANSWER:

- The git pull command first runs git fetch which downloads content from the specified remote repository. Then a git merge is executed to merge the remote content refs and heads into a new local merge commit.

TASK 2 (EXCEPTION HANDLING)

Simple ATM program

Using exception handling code blocks such as try/ except / else / finally, write a program that simulates an ATM machine to withdraw money.

(NB: the more code blocks the better, but try to use at least two key words e.g. try/except):

Tasks:

- Prompt user for a pin code

- If the pin code is correct then proceed to the next step, otherwise ask a user to type in a password again. You can give a user a maximum of 3 attempts and then exit a program.

- Set account balance to 100.

- Now we need to simulate cash withdrawal

- Accept the withdrawal amount

- Subtract the amount from the account balance and display the remaining

balance (NOTE! The balance cannot be negative!)

- However, when a user asks to 'withdraw' more money than they have on their account, then you need to raise an error an exit the program.

ANSWER: python file attached with file name: atmprogram.py

TASK 3 (Testing)

Question 1

Use the Simple
ATM program
to write unit tests for
your functions.

You are allowed to re-factor your function to 'untangle' some logic into smaller

blocks of code to make it easier to write tests.

Try to write at least 5 unit tests in total covering various cases

ANSWER: didnt answer this, I tried one, but I know what format is in my head, I just cant figure out how to write it. I think im bad at testing.