

Test Driven Development

Due Date: April 14 @ 5.00 pm (Pull Request must be generated by this time).

This assignment is intended to familiarize you with Test Driven Development by implementing the User story “Preorder your coffee with a gift card” from Chapter 8 using Test-Driven Development in the language of your choice (Python preferred). Your implementation must include the following core objects: `OrderInformation`, `GiftCard`, and `Receipt`. You must also create the appropriate testing code. Store all code for this project in Stash using Git.

This assignment must be completed using test-driven development. In other words, tests must be written before developing objects. In order to ensure that this occurs, tests must be committed using Git before any code is written. Also, each team member must contribute tests as well as actual code. There must be at least 5 tests for each object.

Rubric: 55 points total as listed below.

- **(15 Points)** Implementation must include the following core classes: `OrderInformation`, `GiftCard`, and `Receipt`.
- **(15 Points)** Test-Driven Development must be used including a minimum of 5 Tests per object. This means that you must develop meaningful tests to test the functionality of each test. *All tests must be in the same file and/or class.*
- **(5 Points)** Fork the repository “Assignment 5” under the project “Assignment #5 – Test-Driven Development”.
- **(5 Points)** Forked repository must be named “Assignment 5 – Team XX” and must be under the project “Assignment #5 – Test Driven Development”.
- **(5 Points)** Each student must create a branch for their own work using their username as the branch name.
- **(10 Points)** All branches must be merged into the main branch before turning in the assignment. While I have covered the basics of merging, I expect you to look into this topic and come to me outside of class with any questions.
- You can self sign-up for your groups via Canvas under People → “Assignment #5 Groups”. Groups must be 2 to 4 people and be created by April 2 at 5 pm. There are no exceptions. Failure to meet this requirement will result in a grade of 0 on this assignment.

Additional Requirements

- Groups must be between 2 and 4 members, no exceptions.
- Each team member must contribute both code and tests.
- Each student must work on their own branch.
- Chelsea Zhang must be assigned as the reviewer when creating your pull request.
- Assignment must be turned in via Pull Request.
- Your code should look professional! Treat this assignment as if you were handing it to a real-world client. You must produce code that is well-formatted, well-aligned, and easy to read.
- *I strongly suggest you use command line only tools for using Git!*

All work is to be version controlled using Stash on voyager (<http://voyager.cs.bgsu.edu:7990/>) under the project “Assignment #5 – Test Driven Development”. The repository to be forked is named “Assignment 5”. You and your group have access to this system using your BGSU ID/Password. The fork must be created under the project “Assignment #5 – Test Driven Development” so that I have access to it.

Commands for Branching/Merging.

Create a branch:	<code>git branch BRANCHNAME</code>
Switch to a branch:	<code>git checkout BRANCHNAME</code>
Create/Switch to a branch:	<code>git checkout -b BRANCHNAME</code>
Switch to master branch:	<code>git checkout master</code>
Merge branch into current branch:	<code>git merge BRANCH_TO_MERGE_INTO_CURRENT_BRANCH</code>

Resources for Test-Driven Development

- <https://docs.python.org/2/library/unittest.html>
- <http://pymotw.com/2/unittest/>
- <http://www.jeffknupp.com/blog/2013/12/09/improve-your-python-understanding-unit-testing/>
- <http://www.drdobbs.com/testing/unit-testing-with-python/240165163>
- <http://code.tutsplus.com/tutorials/the-newbies-guide-to-test-driven-development--net-13835>

Resources for Branching/Merging with git

- <http://pcottle.github.io/learnGitBranching/>
- <https://www.atlassian.com/git/tutorials/using-branches/git-merge>
- <http://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>
- <http://rogerdudler.github.io/git-guide/>
- <http://www.gitguys.com/topics/merging-branches-without-a-conflict/>
- <http://rypress.com/tutorials/git/branches-2>
- <http://nvie.com/posts/a-successful-git-branching-model/>

Tutorials for various git tools:

- Git GUI
 - <http://nathanj.github.io/gitguide/tour.html>
- SourceTree
 - <http://rancoud.com/sourcetree-git-use/>. (Help is also built in to the product.)
- TortoiseGit
 - <https://www.youtube.com/watch?v=pp2S2lHjzZI>
 - <http://robertgreiner.com/2010/02/getting-started-with-git-and-tortoisegit-on-windows/>