

Instructions for First Coding Assignment 1DL610 HT23

Ram Thinniyam

November 12, 2023

1 General Instructions

To begin with, **you must create a git repository on github/gitlab for your group and copy the assignment code into it, giving view access to all the teaching staff (myself + 3TAs)**. You will then implement all the testing inside of this repository and also all the documentation should be uploaded inside it too. **The assignment submission deadline by which time you must push all commits is 1700, 24 November 2023**. The code will be frozen at this time and you will not be allowed to work any further on the assignment. After this, you will have to fix an appointment with the assigned TA to present your work.

Each member of the group will have to present one of the four tasks to the assigned TA during the appointment. For groups with four members, all four tasks have to be covered by the group as a whole. For groups with 3 members, the first three tasks have to be covered. You are allowed to collaborate within your group in order to write up the test code. You are not allowed to collaborate with other groups.

2 Testing

There should be one test script for each function which automatically runs all the test cases associated with that function. Test cases for a particular function must be built using the work flow as for the example `triang` function shown in the lecture. In other words, for each function you must examine the different kinds of inputs:

1. The inputs to the function eg user and cart are the two inputs of `checkout`.
2. Any user input that the function may request.
3. File accesses made by the function.

Then you must do input domain modelling. First identify the valid type of the input. For all functions, create 10 test cases where the input type is valid (except for the `checkoutAndPayment` function, for which you should write 20 cases). Pay special attention to inputs of valid type but which are still invalid eg negative integer input for number of items. Your test cases should be aimed at covering as many equivalence classes as possible. Select one function and for

each invalid input type in $\{int, float, string, list\}$, create a test case. In order to actually run your tests, you will have to create stubs and drivers (for example using fixtures as shown in class). These should ensure that: 1) External files are left unmodified at the end of the tests. 2) The effect of calling other functions by the particular function you are testing must be provided by the stubs.

2.1 Documentation

For each task, you must prepare test documentation. This consists of test function documentation for each function. **Please use the template given for this purpose.** You must also have a short writeup to explain how you went about the input domain modelling for the function. The format of this latter document is left to you, its main purpose is to help you think and communicate what you did to the TA during the presentation.

3 Tasks

Task 1

Modify the login file to implement the following additional functionality in the login file: If the user name is not to be found in the users file, ask if the user would like to register. If yes, then ask user to input a password. Check whether the password has at least 1 capital letter, at least one special symbol and is of length at least 8. If all these checks pass, then add the new user and his password to the user file with initial wallet amount of 0. Test the functions `login()` and `logout(cart)` after the above changes.

Task 2

Test the three functions `display_csv_as_table`, `display_filtered_table` and `searchAndBuyProduct`

Task 3

Test the three functions `load_products_from_csv`, `checkout` and `check_cart`

Task 4

Modify the `checkoutAndPayment` function to update the wallet information in the user file before the message "You have been logged out" is displayed. After this modification, test the `checkoutAndPayment` function.

4 Support

The TAs will have an office hour per week for you to talk to them regarding questions you have about the topics/files they are in charge of as given below:

- **Induni** Files: `login.py`, `logout.py`, `Test Function Document Template-v.2.docx`. Questions related to documentation.
- **Ankur** Files: `product.py`, `products.csv`, `users.json`. Questions regarding stubs and drivers.
- **Prachi** Files: `checkout_and_payment.py`