Homework 2: Ice Cream Stand

(Deadline as per Canvas)

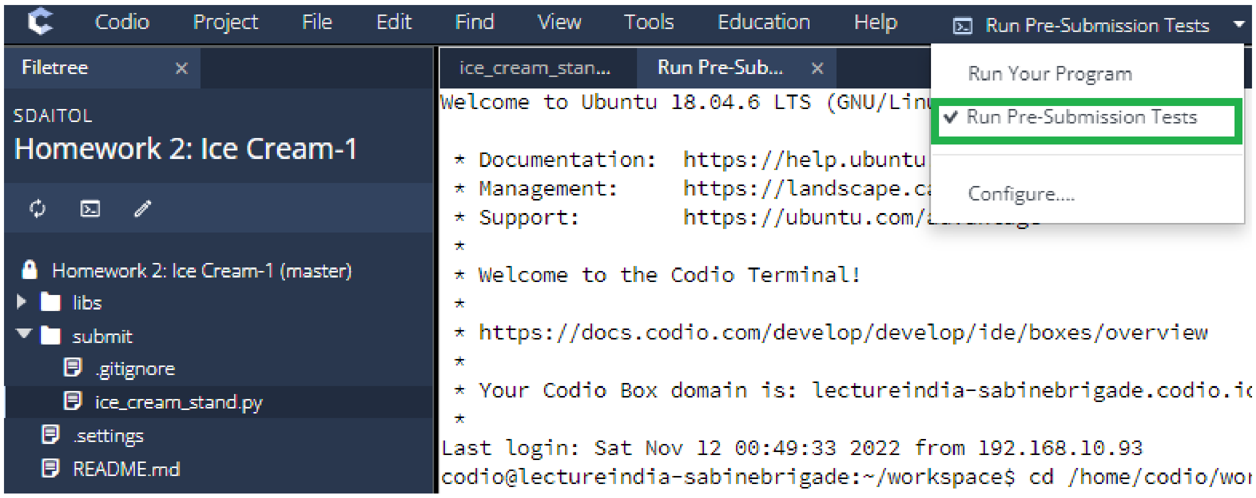
This homework covers concepts learned in Modules 1 and 2.

General Idea of the Assignment

Ice Cream Stand is a program representing taking in customers' ice cream orders and computing the total revenue. You will be processing inputs and calculating the revenue based on the ice cream sizes and quantities.

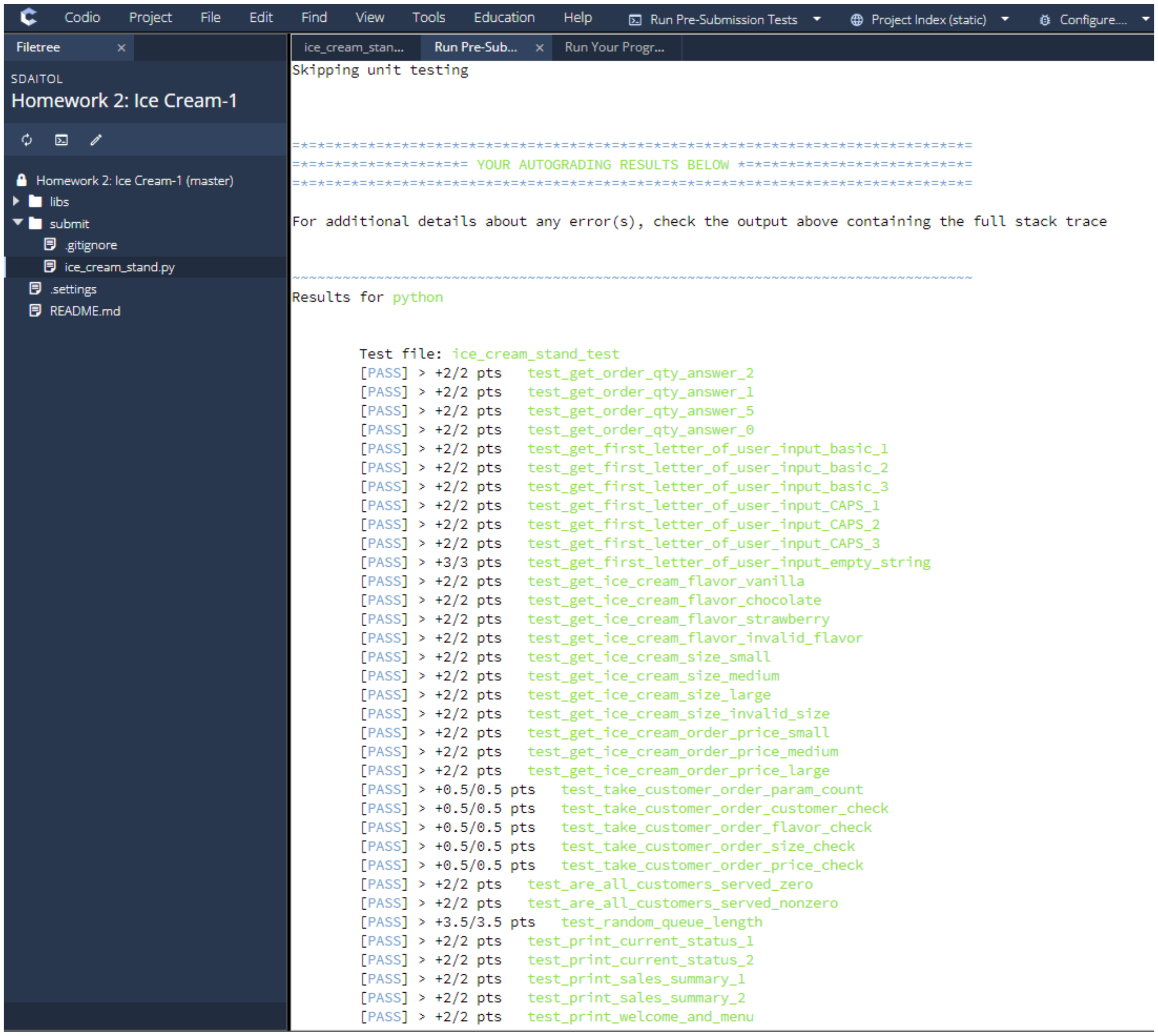
We are providing you with a starter code **ice_cream_stand.py**. The functions have been defined, but just about all of the actual code has been deleted. Your task is to finish the program by adding the necessary code where a **TODO** is indicated. Docstrings and hints are provided and tell you what must be done in each function.

Most of the functionality check will be autograded and is available to you by doing a **"Run Pre-Submission Tests"**.



```
Welcome to Ubuntu 18.04.6 LTS (GNU/Lin

 * Documentation:  https://help.ubuntu
 * Management:     https://landscape.c
 * Support:        https://ubuntu.com/
 *
 * Welcome to the Codio Terminal!
 *
 * https://docs.codio.com/develop/develop/ide/boxes/overview
 *
 * Your Codio Box domain is: lectureindia-sabinebrigade.codio.i
 *
Last login: Sat Nov 12 00:49:33 2022 from 192.168.10.93
codio@lectureindia-sabinebrigade:~/workspace$ cd /home/codio/wo
```

If you run the pre-submission tests initially with just the starter code, you will get failed tests but once you have filled in and implemented the functions correctly, the results would look like:

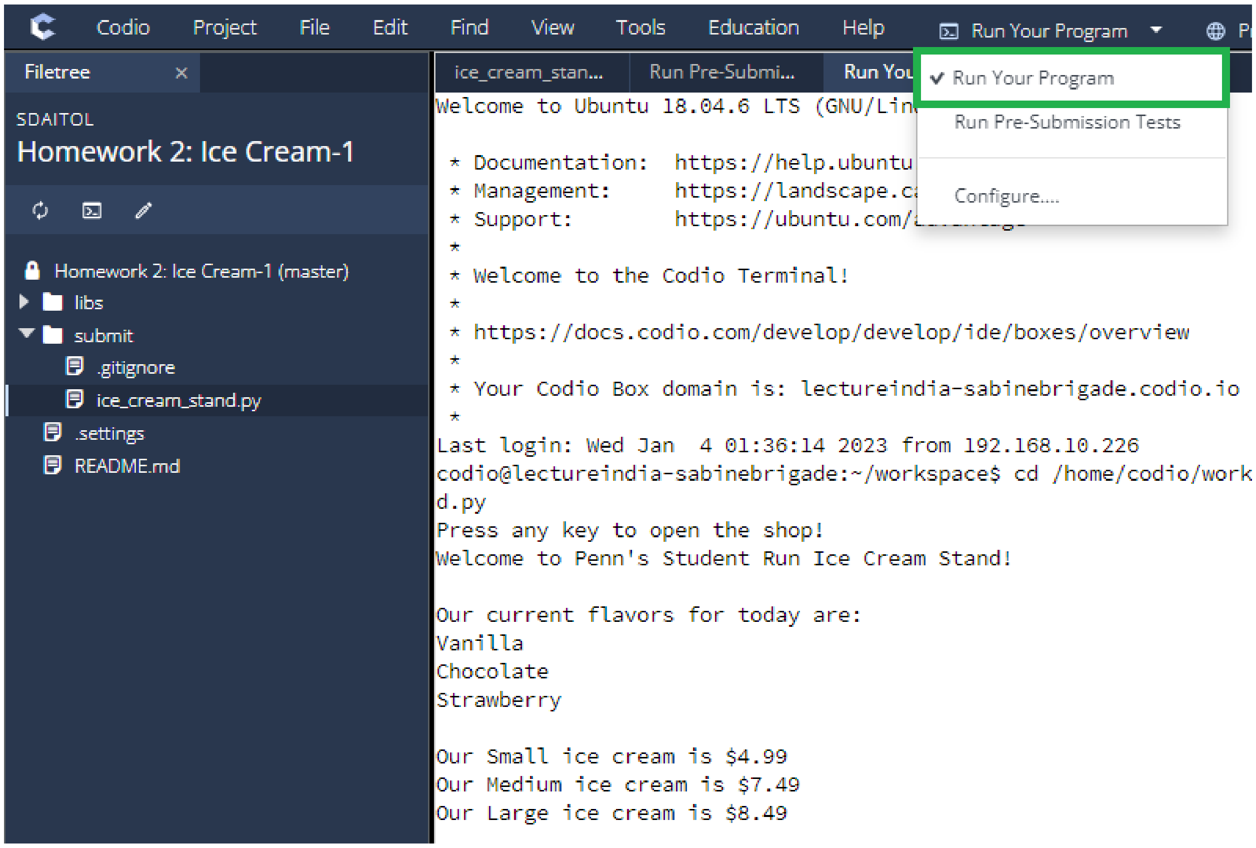


```
Skipping unit testing


=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=
=*=*=*=*=*=*=*=*=*=*= YOUR AUTOGRADING RESULTS BELOW *=*=*=*=*=*=*=*=*=*=*=*=*=
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=

For additional details about any error(s), check the output above containing the full stack trace


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Results for python


        Test file: ice_cream_stand_test
        [PASS] > +2/2 pts   test_get_order_qty_answer_2
        [PASS] > +2/2 pts   test_get_order_qty_answer_1
        [PASS] > +2/2 pts   test_get_order_qty_answer_5
        [PASS] > +2/2 pts   test_get_order_qty_answer_0
        [PASS] > +2/2 pts   test_get_first_letter_of_user_input_basic_1
        [PASS] > +2/2 pts   test_get_first_letter_of_user_input_basic_2
        [PASS] > +2/2 pts   test_get_first_letter_of_user_input_basic_3
        [PASS] > +2/2 pts   test_get_first_letter_of_user_input_CAPS_1
        [PASS] > +2/2 pts   test_get_first_letter_of_user_input_CAPS_2
        [PASS] > +2/2 pts   test_get_first_letter_of_user_input_CAPS_3
        [PASS] > +3/3 pts   test_get_first_letter_of_user_input_empty_string
        [PASS] > +2/2 pts   test_get_ice_cream_flavor_vanilla
        [PASS] > +2/2 pts   test_get_ice_cream_flavor_chocolate
        [PASS] > +2/2 pts   test_get_ice_cream_flavor_strawberry
        [PASS] > +2/2 pts   test_get_ice_cream_flavor_invalid_flavor
        [PASS] > +2/2 pts   test_get_ice_cream_size_small
        [PASS] > +2/2 pts   test_get_ice_cream_size_medium
        [PASS] > +2/2 pts   test_get_ice_cream_size_large
        [PASS] > +2/2 pts   test_get_ice_cream_size_invalid_size
        [PASS] > +2/2 pts   test_get_ice_cream_order_price_small
        [PASS] > +2/2 pts   test_get_ice_cream_order_price_medium
        [PASS] > +2/2 pts   test_get_ice_cream_order_price_large
        [PASS] > +0.5/0.5 pts   test_take_customer_order_param_count
        [PASS] > +0.5/0.5 pts   test_take_customer_order_customer_check
        [PASS] > +0.5/0.5 pts   test_take_customer_order_flavor_check
        [PASS] > +0.5/0.5 pts   test_take_customer_order_size_check
        [PASS] > +0.5/0.5 pts   test_take_customer_order_price_check
        [PASS] > +2/2 pts   test_are_all_customers_served_zero
        [PASS] > +2/2 pts   test_are_all_customers_served_nonzero
        [PASS] > +3.5/3.5 pts   test_random_queue_length
        [PASS] > +2/2 pts   test_print_current_status_1
        [PASS] > +2/2 pts   test_print_current_status_2
        [PASS] > +2/2 pts   test_print_sales_summary_1
        [PASS] > +2/2 pts   test_print_sales_summary_2
        [PASS] > +2/2 pts   test_print_welcome_and_menu
```

For the autograder to run properly, do not change the function names or the parameters. Do not also add optional parameters and change the return types. The ice_cream_stand.py should also remain in the submit folder and should not be renamed otherwise the autograder will fail to locate your program.

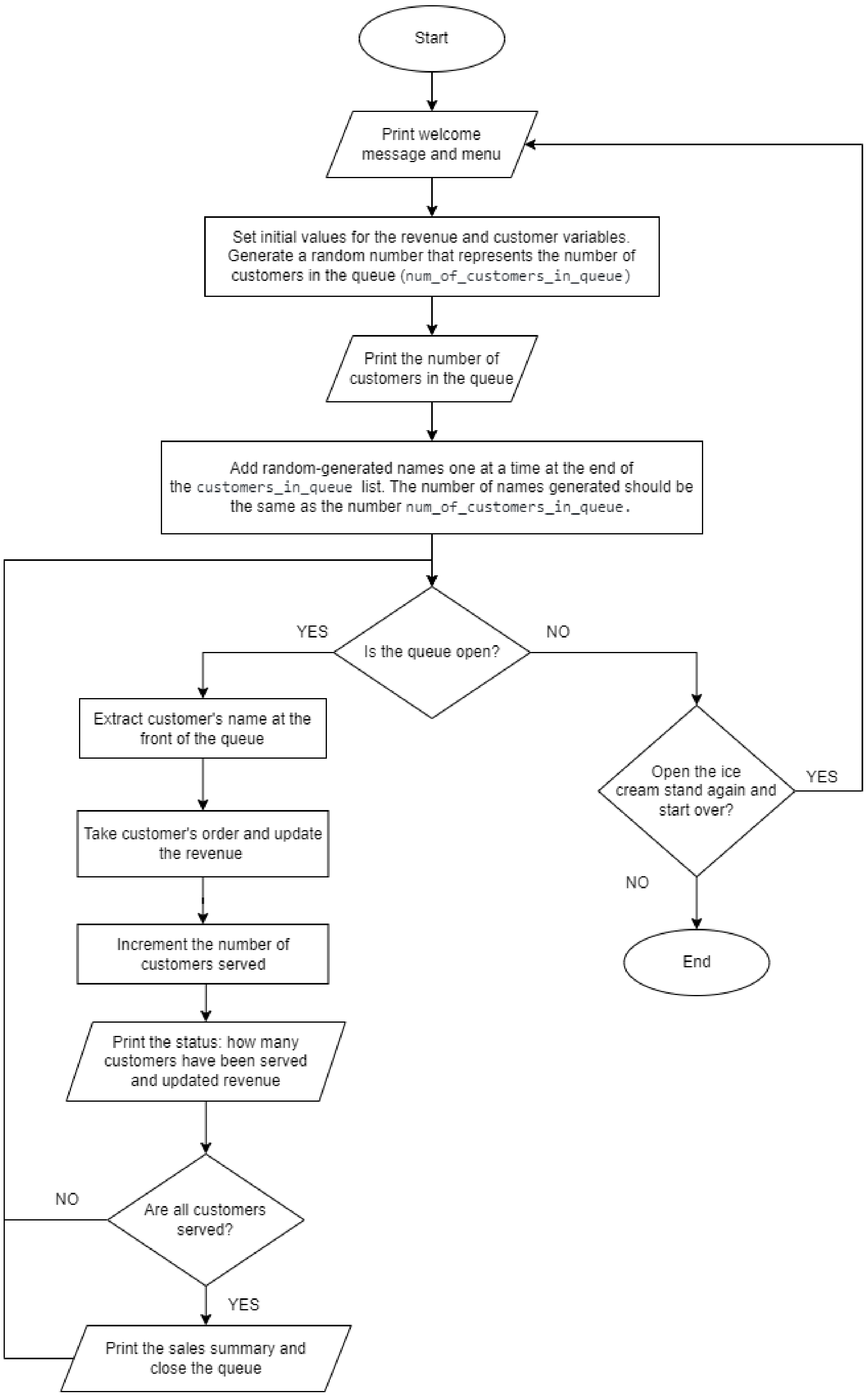
Program Output

We have provided a *template_behavior.txt* file which shows some sample runs of the program -- yours should provide similar information. You can test this using the **"Run Your Program".**



```
Welcome to Ubuntu 18.04.6 LTS (GNU/Lin

 * Documentation:  https://help.ubuntu
 * Management:     https://landscape.c
 * Support:        https://ubuntu.com/
 *
 * Welcome to the Codio Terminal!
 *
 * https://docs.codio.com/develop/develop/ide/boxes/overview
 *
 * Your Codio Box domain is: lectureindia-sabinebrigade.codio.io
 *
Last login: Wed Jan  4 01:36:14 2023 from 192.168.10.226
codio@lectureindia-sabinebrigade:~/workspace$ cd /home/codio/work
d.py
Press any key to open the shop!
Welcome to Penn's Student Run Ice Cream Stand!

Our current flavors for today are:
Vanilla
Chocolate
Strawberry

Our Small ice cream is $4.99
Our Medium ice cream is $7.49
Our Large ice cream is $8.49
```

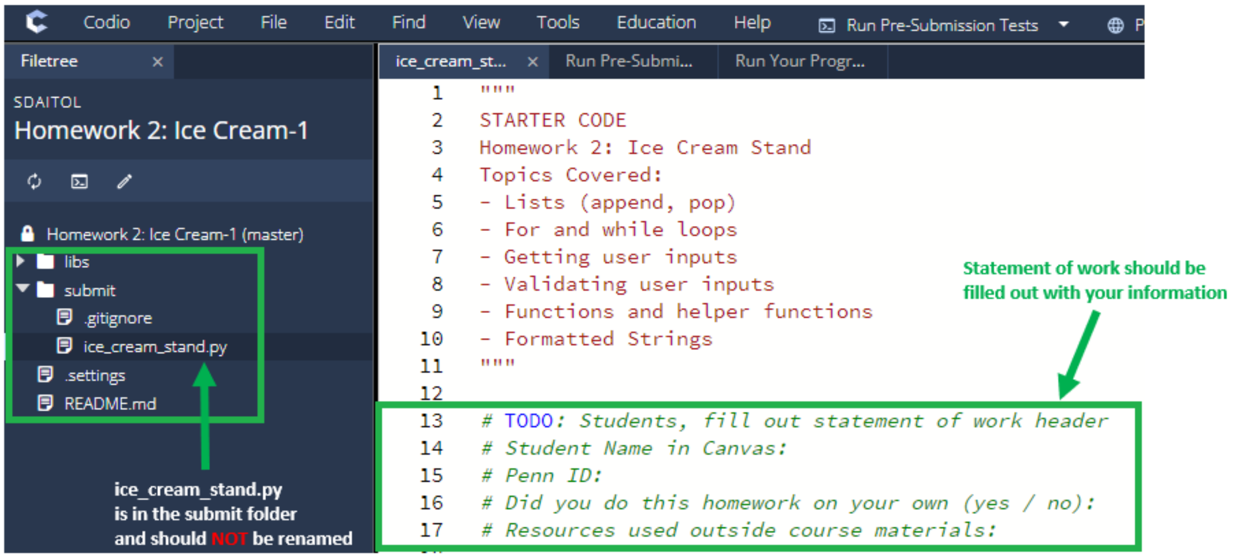
Your program is also expected to follow the flowchart shown on the next page.


Start
Print welcome message and menu
Set initial values for the revenue and customer variables. Generate a random number that represents the number of customers in the queue (num_of_customers_in_queue)
Print the number of customers in the queue
Add random-generated names one at a time at the end of the customers_in_queue list. The number of names generated should be the same as the number num_of_customers_in_queue.
Is the queue open?
YES
NO
Extract customer's name at the front of the queue
Take customer's order and update the revenue
Increment the number of customers served
Print the status: how many customers have been served and updated revenue
Are all customers served?
NO
YES
Print the sales summary and close the queue
Open the ice cream stand again and start over?
YES
NO
End

Submission

Your submission should include:

- *ice_cream_stand.py* - the source code for your program
- Fill out the statement of work header
- In Codio, your file structure should look like the screenshot below:

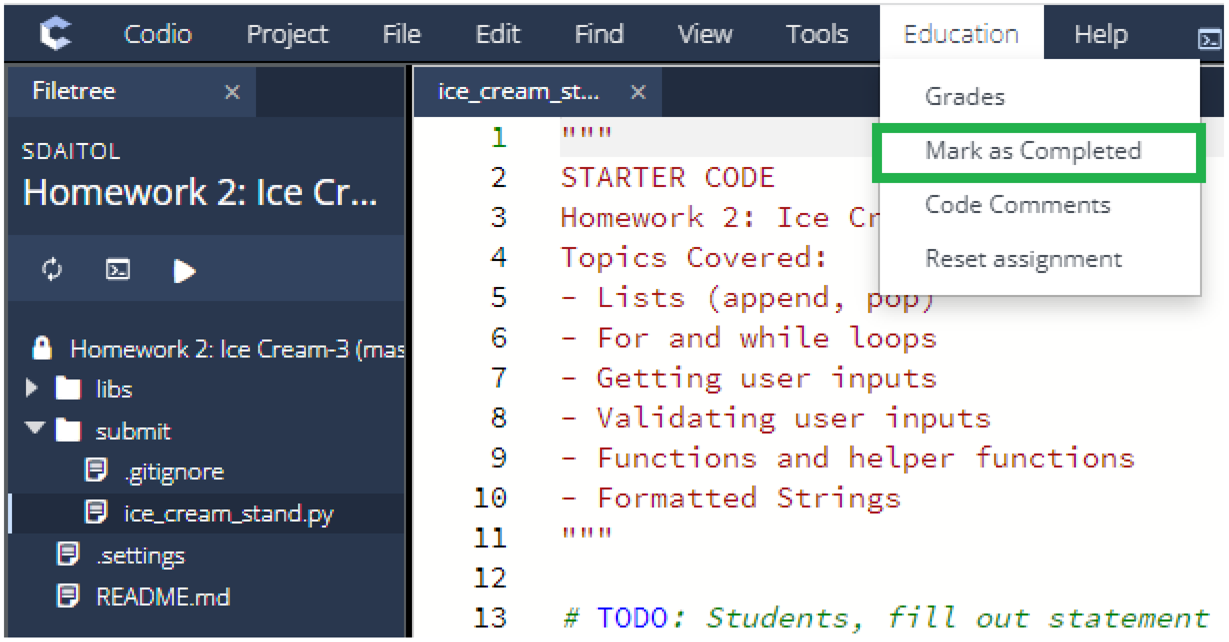


```
"""
STARTER CODE
Homework 2: Ice Cream Stand
Topics Covered:
- Lists (append, pop)
- For and while loops
- Getting user inputs
- Validating user inputs
- Functions and helper functions
- Formatted Strings
"""

# TODO: Students, fill out statement of work header
# Student Name in Canvas:
# Penn ID:
# Did you do this homework on your own (yes / no):
# Resources used outside course materials:
```

Statement of work should be filled out with your information

ice_cream_stand.py is in the submit folder and should NOT be renamed

- If you are ready to submit your homework, mark your submission as completed.



```
"""
STARTER CODE
Homework 2: Ice Cr
Topics Covered:
- Lists (append, pop)
- For and while loops
- Getting user inputs
- Validating user inputs
- Functions and helper functions
- Formatted Strings
"""

# TODO: Students, fill out statement
```

Evaluation (Total 100 points)

- **Program Correctness (Fully Autograded) - 65 pts**
  These tests are available to you by doing a "Run Pre-Submission Tests". If there are indentation errors, syntax errors, or incorrect file name and function names and signatures, the autograder will fail to run. Please read the error messages and resolve them otherwise you will get a zero in the autograded portion.

- **Correctness in the take_customer_order function (Manually graded)- 10 pts**
  These are the requirements as stated in the TODO statements in the starter code.

- **Correctness of the main method (Manually graded) - 13 pts**
  These are the requirements as stated in the TODO statements in the starter code.

- **User interface (Manually graded) - 2 pts**
  This includes readability when you are printing information statements such as the welcome and status messages and readability when you are asking for user input. When asking input from a user, are you printing the allowable choices accepted by your program? Since we are running the program only in the console, will the user be able to easily differentiate the orders taken in per customer (can be achieved using sufficient white spaces)? As a general rule, will your user interface be easily understandable by a potential user of this program?

- **Code setup and style (Manually graded) - 10 pts**
  Code setup means that your code runs properly in Codio - no indentation errors, syntax errors, file is correctly named and is in the correct submit folder or other errors preventing the program from successfully running. Style includes descriptive variable names and should be based on the information they store. If you use helper functions, they should also be named descriptively and named based on what they do and should be properly commented and docstrings should be added. Style also includes adding in comments for non-trivial lines of code. There will be no penalty for over-commenting so if a particular line of code needs to be explained, especially if it is doing some calculations or important flow control changes in your program, then put in comments on what that line of code is supposed to do. Use the provided starter comments in the take_customer_order and main functions as a guide on what we expect a sufficient amount of code commenting looks like.