

## Homework 3: Ice Cream Stand

HW deadline as per Canvas.

This assignment deals with the following topics:

- Lists (append, pop)
- For and while loops
- Getting user inputs
- Validating user inputs
- Functions and modular programming
- Formatted Strings

### General Idea of the Assignment

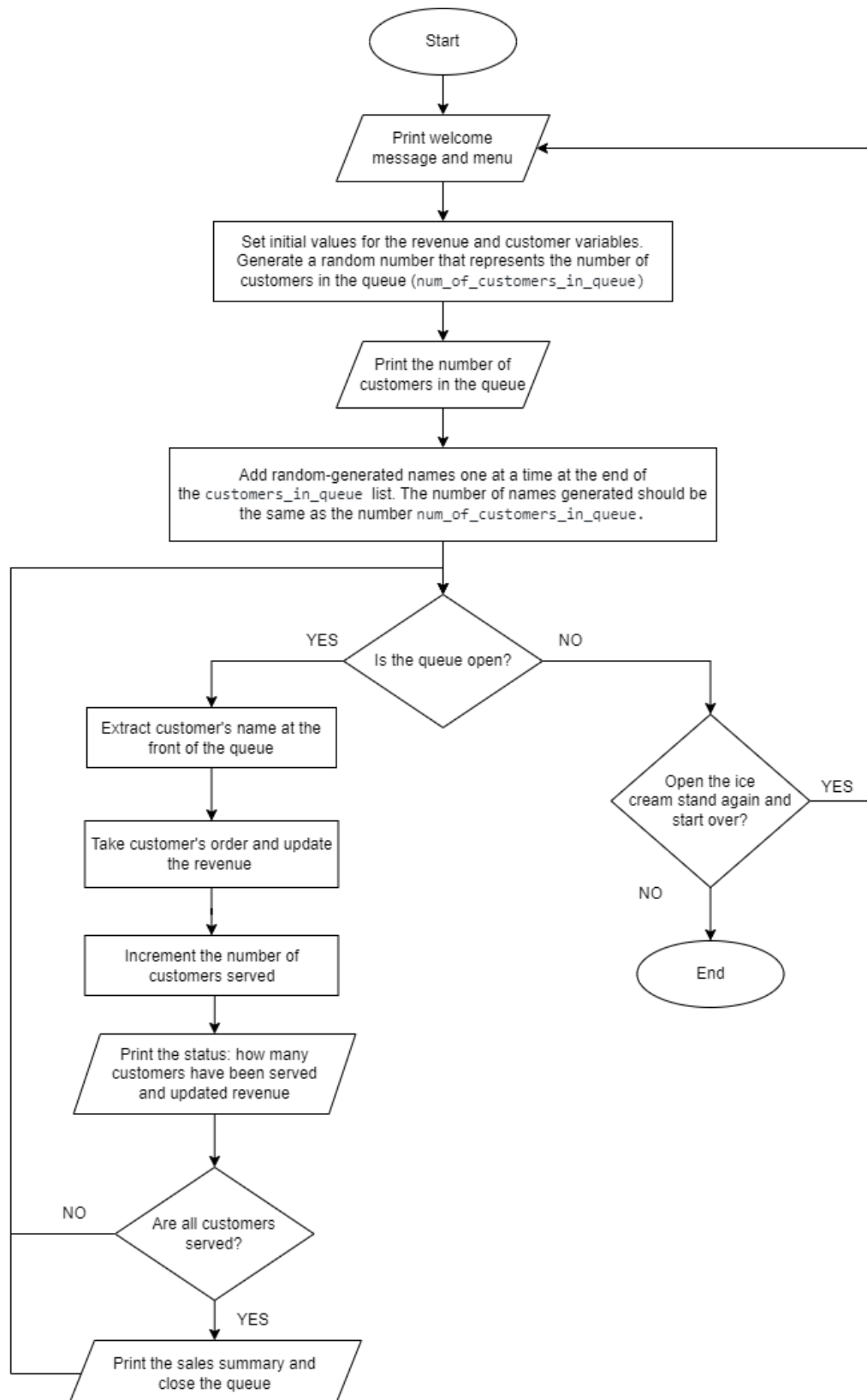
Ice Cream Stand is a program representing taking in customers' ice cream orders and computing the total revenue. You will be processing inputs and calculating the revenue based on the ice cream sizes and quantities.

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.

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



## **Submission**

Your submission should include:

- *ice\_cream\_stand.py* - the source code for your program
- Fill out the statement of work header

## **Evaluation**

- Program Correctness -15 pts  
Does the program work as expected? Did you follow the directions exactly? Did you implement all the required functions correctly? Does your program properly handle user input? For example, when the user is asked to “input the number of ice creams you want”, will the program behave as expected for different inputs like 5 or 2.3 or ‘abcd’? Another example, is the `get_first_letter_of_user_input` function, does it return the value as expected?
- Correctness in the `take_customer_order` function - 5 pts  
These are the requirements as stated in the TODO statements in the starter code.
- Correctness of the main method - 5 pts  
These are the requirements as stated in the TODO statements in the starter code.
- User interface - 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. As a general rule, will your user interface be easily understandable by a potential user of this program?
- Code setup and style - 3 pts  
Code setup means that your code runs properly. 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. 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.