

You can watch it work here:

<http://www.csit.parkland.edu/~kurban/permanent/labs/stackqueue/runme.cgi>

# Stacks and Queues

servers and customers

# Requirements

Create file in python with a **comment** containing the academic honesty pledge as shown below. Add another, separate comment to the file containing your name

- Write a python program that prints the solution to the servers & customers supplied in a textareas as described on a later slide.
- Use a separate stack class and a separate queue class.
- Attach your python source in using the dropbox link in cobra learning. (I don't need any html files)
- **Put a link to your running file in the message area**

```
# I honor Parkland's core values by affirming that I have  
# followed all academic integrity guidelines for this work.  
  
# your name
```

# Servers and customers

Servers are stored in a stack, where the first server in the textarea is the the last server used. **It is a requirement that you use a stack of servers.**

Customers are stored in a queue, where the first customer in the textarea is the first customer used. **It is a requirement that you use a queue of customers.**

Each server and customer has a Name and a Number associated with it. The Name is used for printing and the Number is the amount the server can serve or the customer needs served.

Each will have a separate text area.

# What to do (inside a loop)

"load" the first server (at the bottom) and the first customer (at the top):

- If they are the same, report that and remove both from their respective areas and move on to the next two.
- If the server can serve more than the customer requires, the server completely serves the customer and reduces the Number on that server by the amount served. Remove the customer and move on to the next customer.
- If the customer needs more served than the server can serve, the customer completely uses up the server and reduces the Number on that customer by the amount served. Remove the server and move on to the next server.

# When to end (the loop condition)

- If all the servers are done serving, report that there are no more servers.
- If all the customers are done being served, report that there are no more customers.
- ~~There might be a tie, so both should be reported.~~ (Edit: don't worry about this - Ken)

# hints & ideas

- make a form with two text areas and one submit. A sample of the form is [here](#).
- make a class that contains the name and number, then make queues and stacks of that class. (Of course you can use a list instead of the class)
- Plan:
  - start with a form that reads and prints the data
  - put the servers in a stack and print them
  - put the customers in a queue and print them
  - work on the top/first <, ==, >
- <https://repl.it/@kurbanParkland/splittingstuff> has the logic for splitting up the into to a usable format