Problem Wk.2.2: Nano Quiz

This problem has been submitted previously; no changes will be saved.

Due date: 2/10, 9:50am

Do all parts of this problem and then click Submit. There is only one Submit button, you should only do that when you are finished with all the parts of the problem.

Do not try to start another log in, you will lose what you typed.

There is a limited checking budget for this problem (10 checks).

You have 15 minutes. You must click submit before:

2/10, 9:50am

Part 1: Hammock

Define a class `Hammock` that keeps track of who is allowed to sit on a hammock. If the hammock is empty, then anyone can sit on it. If the hammock is not empty, then a new request to sit on it will be refused. However, if the next request to sit is from the same person who was refused last time, then the request is granted (as a reward for their persistence).

The `Hammock` should support three operations:

- `__init__` initializes the hammock -- it starts out empty.
- `sitDown`: takes the name of a person that sits down on the hammock. if they successfully sit down, it returns `'welcome!'` otherwise, it returns `'sorry, no room'`
- `leave`: someone leaves the hammock -- returns the number of people that are left. if nobody was on the hammock, return 0

You do not have to handle illegal actions, such as a sitDown request from someone already in the hammock.

Here's an example transcript:

```
>>> myHammock = Hammock()
>>> myHammock.sitDown('George')
welcome!
>>> myHammock.sitDown('Bobby')
sorry, no room
>>> myHammock.sitDown('Bobby')
welcome!
>>> myHammock.leave()
1
>>> myHammock.leave()
0
>>> myHammock.leave()
0
>>> myHammock.sitDown('Martha')
welcome!
>>> myHammock.sitDown('Wilhelm')
sorry, no room
>>> myHammock.sitDown('Klaus')
sorry, no room
```

```
>>> myHammock.sitDown('Wilhelm')
sorry, no room
>>> myHammock.leave()
0
```

```
class Hammock:
    def __init__(self):
        self.inHammock = 0
        self.lastRefused = None
    def sitDown(self, name):
        if self.inHammock==0 or name==self.lastRefused:
            self.inHammock+=1
            return 'welcome!'
        self.lastRefused=name
        return 'sorry, no room'
    def leave(self):
        if self.inHammock>0:
            self.inHammock-=1
        return self.inHammock
```

This is the answer we wrote:

```
class Hammock:
    def __init__(self):
        self.occupants = 0
        self.requester = None
    def sitDown(self, name):
        if self.occupants == 0:
            self.occupants += 1
            return 'welcome!'
        elif name == self.requester:
            self.occupants += 1
            self.requester = None
            return 'welcome!'
        else:
            self.requester = name
            return 'sorry, no room'
    def leave(self):
        if self.occupants > 0:
            self.occupants -= 1
            return self.occupants
        else:
            return 0
```

8 checks left Check

Part 2: Enable Submit

Part 2: Enable Submit

Current time is: **3/1/2011, 9:09pm**

Click Submit before: **2/10, 9:50am**

The Check button will update the current time.

Enter Done below

[ ] 

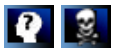
and click Submit.

If this problem is submitted past the due time, this subproblem will be marked incorrect.

8 checks left Check

This is a multi-part problem, each part has its own Save and Check buttons but there is ONLY ONE Submit button for the WHOLE problem. Finish working on all the parts before you click on Submit.

Get Answers



Copyright © 2003 by Massachusetts Institute of Technology. All rights reserved. MIT
Send comments or questions to 6.01 On Line

MIT OpenCourseWare
http://ocw.mit.edu

6.01SC Introduction to Electrical Engineering and Computer Science

Spring 2011

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.