Outline

1. Queues
   - What Is a Queue?
   - The Queue Abstract Data Type
   - Implementing a Queue in Python
   - Simulation: Hot Potato
   - Simulation: Printing Tasks

2. Deque
   - What Is a Deque?
   - The Deque Abstract Data Type
   - Implementing a Deque in Python
   - Palindrome-Checker
Outline

1 Queues

- What Is a Queue?
  - The Queue Abstract Data Type
  - Implementing a Queue in Python
  - Simulation: Hot Potato
  - Simulation: Printing Tasks

2 Deque

- What Is a Deque?
  - The Deque Abstract Data Type
  - Implementing a Deque in Python
  - Palindrome-Checker
A Queue of Python Data Objects

```
<table>
<thead>
<tr>
<th>rear</th>
<th>8.4</th>
<th>True</th>
<th>&quot;dog&quot;</th>
<th>4</th>
<th>front</th>
</tr>
</thead>
</table>
```

items
Outline

1 Queues
   - What Is a Queue?
   - The Queue Abstract Data Type
   - Implementing a Queue in Python
   - Simulation: Hot Potato
   - Simulation: Printing Tasks

2 Deque
   - What Is a Deque?
   - The Deque Abstract Data Type
   - Implementing a Deque in Python
   - Palindrome-Checker
Queue() creates a new queue that is empty. It needs no parameters and returns an empty queue.

enqueue(item) adds a new item to the rear of the queue. It needs the item and returns nothing.

dequeue() removes the front item from the queue. It needs no parameters and returns the item. The queue is modified.

isEmpty() tests to see whether the queue is empty. It needs no parameters and returns a boolean value.

size() returns the number of items in the queue. It needs no parameters and returns an integer.
Queues

What Is a Queue?
The Queue Abstract Data Type
Implementing a Queue in Python
Simulation: Hot Potato
Simulation: Printing Tasks

Deque

What Is a Deque?
The Deque Abstract Data Type
Implementing a Deque in Python
Palindrome-Checker
Queue Implementation in Python

```python
class Queue:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def enqueue(self, item):
        self.items.insert(0, item)

    def dequeue(self):
        return self.items.pop()

    def size(self):
        return len(self.items)
```

Outline

1. Queues
   - What Is a Queue?
   - The Queue Abstract Data Type
   - Implementing a Queue in Python
   - Simulation: Hot Potato
   - Simulation: Printing Tasks

2. Deque
   - What Is a Deque?
   - The Deque Abstract Data Type
   - Implementing a Deque in Python
   - Palindrome-Checker
A Six Person Game of Hot Potato

Bill
David
Susan
Jane
Kent
pass to next person

After 5 passes, Brad is eliminated

Until predefined counting constant

Brad
Kent

And so on

Susan
Jane

Basic Data Structures
A Queue Implementation of Hot Potato

Brad Susan David Bill front rear
Go to the rear
(Pass the potato)
Jane Kent
Bill Susan David front rear Jane Kent Brad
dequeue enqueue

Basic Data Structures
def hotPotato(namelist, N):
    simqueue = Queue()
    for name in namelist:
        simqueue.enqueue(name)

    while simqueue.size() > 1:
        for i in range(N):
            simqueue.enqueue(simqueue.dequeue())

    simqueue.dequeue()

    return simqueue.dequeue()
Outline

1. Queues
   - What Is a Queue?
   - The Queue Abstract Data Type
   - Implementing a Queue in Python
   - Simulation: Hot Potato
   - Simulation: Printing Tasks

2. Deque
   - What Is a Deque?
   - The Deque Abstract Data Type
   - Implementing a Deque in Python
   - Palindrome-Checker
Printer Queue Simulation–The Printer Class

```python
class Printer:
    def __init__(self, pages):
        self.pagerate = pages
        self.currentTask = None
        self.timeRemaining = 0

    def tick(self):
        if self.currentTask != None:
            self.timeRemaining = self.timeRemaining - 1
            if self.timeRemaining == 0:
                self.currentTask = None
```

Printer Queue Simulation–The Printer Class II

```python
16     def busy(self):
17         if self.currentTask != None:
18             return True
19         else:
20             return False
21
22     def startNext(self,newtask):
23         self.currentTask = newtask
24         self.timeRemaining = newtask.getPages() \
25             * 60/self.pagerate
```
import random

class Task:
    def __init__(self, time):
        self.timestamp = time
        self.pages = random.randrange(1, 21)

    def getStamp(self):
        return self.timestamp

    def getPages(self):
        return self.pages

    def waitTime(self, currenttime):
        return currenttime - self.timestamp
Printer Queue Simulation–The Main Simulation

```python
from queue import *
from printer import *
from task import *

import random

import random

def simulation(numSeconds, pagesPerMinute):
    labprinter = Printer(pagesPerMinute)
    printQueue = Queue()
   (waitingtimes = []

    for currentSecond in range(numSeconds):

```

Printer Queue Simulation–The Main Simulation II

```python
if newPrintTask():
    task = Task(currentSecond)
    printQueue.enqueue(task)

if (not labprinter.busy()) and (not printQueue.isEmpty()):
    nexttask = printQueue.dequeue()
    waitingtimes.append(nexttask.waitTime(currentSecond))
    labprinter.startNext(nexttask)

labprinter.tick()

averageWait = sum(waitingtimes) / float(len(waitingtimes))
print "Average Wait Time\%6.2f seconds" % (averageWait),
print "Tasks Remaining \%3d" % (printQueue.size())
```
Printer Queue Simulation–The Main Simulation III

```python
32
33
def newPrintTask():
35    num = random.randrange(1, 181)
36    if num == 180:
37        return True
38    else:
39        return False
```
What if enrollment goes up and the average number of students increases by 20?

What if it is Saturday and students are not needing to get to class? Can they afford to wait?

What if the size of the average print task decreases since Python is such a powerful language and programs tend to be much shorter?
Outline

1 Queues
   • What Is a Queue?
   • The Queue Abstract Data Type
   • Implementing a Queue in Python
   • Simulation: Hot Potato
   • Simulation: Printing Tasks

2 Deque
   • What Is a Deque?
   • The Deque Abstract Data Type
   • Implementing a Deque in Python
   • Palindrome-Checker
A Deque of Python Data Objects

```
items
"dog" 4 "cat" True
```

```
add to rear
rear
front
add to front
```

remove from rear
remove from front
Outline

1. Queues
   - What Is a Queue?
   - The Queue Abstract Data Type
   - Implementing a Queue in Python
   - Simulation: Hot Potato
   - Simulation: Printing Tasks

2. Deque
   - What Is a Deque?
   - The Deque Abstract Data Type
   - Implementing a Deque in Python
   - Palindrome-Checker
Deque() creates a new deque that is empty. It needs no parameters and returns an empty deque.
addFront(item) adds a new item to the front of the deque. It needs the item and returns nothing.
addRear(item) adds a new item to the rear of the deque. It needs the item and returns nothing.
removeFront() removes the front item from the deque. It needs no parameters and returns the item. The deque is modified.
removeRear() removes the rear item from the deque. It needs no parameters and returns the item. The deque is modified.
isEmpty() tests to see whether the deque is empty. It needs no parameters and returns a boolean value.
size() returns the number of items in the deque. It needs no parameters and returns an integer.
Outline

1. Queues
   - What Is a Queue?
   - The Queue Abstract Data Type
   - Implementing a Queue in Python
   - Simulation: Hot Potato
   - Simulation: Printing Tasks

2. Deque
   - What Is a Deque?
   - The Deque Abstract Data Type
   - Implementing a Deque in Python
   - Palindrome-Checker
class Deque:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def addFront(self, item):
        self.items.append(item)

    def addRear(self, item):
        self.items.insert(0, item)

    def removeFront(self):
        return self.items.pop()
Deque Implementation in Python II

```python
16 17  def removeRear(self):
18 19    return self.items.pop(0)
20 21  def size(self):
22 23    return len(self.items)
```
Outline

1. Queues
   - What Is a Queue?
   - The Queue Abstract Data Type
   - Implementing a Queue in Python
   - Simulation: Hot Potato
   - Simulation: Printing Tasks

2. Deque
   - What Is a Deque?
   - The Deque Abstract Data Type
   - Implementing a Deque in Python
   - Palindrome-Checker

Basic Data Structures
A Deque

Add "radar" to the rear

remove from rear

Remove from front and rear
Palindrome Checker

def palchecker(aString):
    chardeque = Deque()

    for ch in aString:
        chardeque.addRear(ch)

    stillEqual = True

    while chardeque.size() > 1 and stillEqual:
        first = chardeque.removeFront()
        last = chardeque.removeRear()
        if first != last:
            stillEqual = False

    return stillEqual