CompSci 101: Test 2 PRACTISE

Peter Lorensen
April 8, 2013

Name: __________________________________________

NetID/Login: __________________________

Community Standard Acknowledgment (signature): ________________________________

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>28 pts</td>
<td></td>
</tr>
<tr>
<td>Problem 2</td>
<td>22 pts</td>
<td></td>
</tr>
<tr>
<td>TOTAL:</td>
<td>50 pts</td>
<td></td>
</tr>
</tbody>
</table>

In writing code you do not need to worry about specifying the proper import statements. Don’t worry about getting function or method names exactly right. Assume that all libraries and packages we’ve discussed are imported in any code you write.
PROBLEM 1: (PAINT CUSTOMERS 28 POINTS)

You will be asked to write code that refers to the list below. The python code you write should work with any values stored in the list.

Given the following data from a painting company:

```python
paintData = [ ["Blue", 35, ("Axson", True)],
              ["Red", 20, ("Jack", True)],
              ["Blue", 35, ("Winy", False)],
              ["Yellow", 45, ("Winy", False)],
              ["Purple", 65, ("Hopi", False)],
              ["Blue", 35, ("Axson", True)],
              ["Yellow", 45, ("Don", True)],
              ["Red", 20, ("Don", True)],
```

It contains:  
```
[ ["Color code", price, (username, premium customer) ] ]
```

With type:  
```
[ [ string, float, ( string, boolean ) ] ]
```

Part A (8 points)

Write code that return a list of tuples like this:

```python
[ ("Axson", True), ("Jack", True), ("Winy", False), ("Winy", False), ("Hopi", False),
  ("Axson", True), ("Don", True), ("Don", True)]
```

The list contains all user tuples from the paintData list above.

The method should return the user information from the paintData list (username, premium customer):

```python
def getUserInfo( paintData ):
```

```python
```
Part B (8 points)

Write code to return a dictionary like this:

```python
{ "Blue": ["Axson"], "Red": ["Jack", "Don"], "Yellow": ["Winy", "Don"], "Purple": ["Hopi"] }
```

The method should return a dictionary of all the paints with the usernames of who has bought it:

```python
def getPaintInfo( paintData ):
```

Part C (12 points)

Write code to find the total amount of money that each user has spent. The function must return a dictionary like this:

```python
{ "Axson": 70, "Jack": 20, "Don": 65, "Winy": 80, "Hopi": 65 }
```

```python
def getUserTotal( paintData ):
PROBLEM 2 : (SHOPPING TOO MUCH 22 POINTS)

You will be asked to write code that refers to the list below. The python code you write should work with any values stored in the list.

An electronic store has a huge sale. Many customers flock to the store to buy cheap TVs, computers, phones, printers etc.

People may only buy 2 of the same item per person. The store intends to enforce this by deducting the extra amount from customers credit cards if they buy more. Some customers, however, try to cheat by going to one of the other stores in town and buy items there.

The electronic store looks over its list of credit card purchases at the end of the day and want to pick out those customers for closer scrutiny who bought items in more than one place. Here is their list of stores and in each store a list of credit cards:

```python
allStores = [['1024 2367 5693 1287", "5615 7245 9813 1267", "1434 8386 6587 8865"],
["1024 2367 5693 1287", "8799 6587 1265 6533"],
["4422 6535 4544 6658", "5615 7245 9813 1267", "9855 6544 1294 3798", "1434 8386 6587 8865"],
["7766 3565 9445 3256", "1434 8386 6587 8865", "5615 7245 9813 1267" ]
```

Part A (14 points)

Write code for the method that return a list of customer credit cards who bought items at more than one store. For the example above the method should return this list:

```python
["1024 2367 5693 1287", "5615 7245 9813 1267", "1434 8386 6587 8865" ]
```

```python
def getPotentialCheats( allStores ):
```
Part B (8 points)

You may use the method written in Part A and assume that it works correctly. Write code for this method that returns a list of tuples that contains the following:

[ (index of store in allStores, number of cust visiting more stores, number of cust visiting ONLY this store) ]

For the example allStores above the method should return this list:

[ (0, 3, 0), (1, 1, 1), (2, 2, 2), (3, 2, 1) ]

def getCustomers( allStores ):
(Extra page for writing code)