CSCE 110: Programming I

Exam #2 — Answer Key

November 4, 2011

Your final score is circled in orange. Below are the solutions for Versions A and B of Exam #2 along with the regrading policy.

1 Regrading Policy

If there are any grading errors related to your exam, you must notify me in writing by Monday, November 14, 2011. After November 14th, no changes to exam grades will be considered. Below, are the steps that you must follow if you want your exam regraded.

1. Write a formal statement that specifies clearly the error in question.

2. Attach your statement to your exam.

3. During class or office hours, give me your statement along with your exam to reconsider.

Finally, if your grading error is related to wanting to receive more partial credit (on question 7 for example), then your exam will be returned back to you. However, if there is an actual error (e.g., a correct solution is marked incorrect, your exam score is not tallied correctly), then please follow the above steps to have your exam regraded.

2 Version A

1.
   a) 16
   b) 65
   c) 6
   d) 20
   e) -65

2.
   a) def, return, print
   b) 1, 12, 7, 8, 9, 3, 4, 5, 9, 10, 12, 13
   c) The value of result is 25.
   d) n
e) NONE

3.

a) 2
b) 2
c) set(['tulip', 'lily', 7]) # set elements can be in any order
d) set([7])
e) set(['lily'])

4.

a) [1, 2, 3, 4, 5]

5.

a) float
b) string
c) 4.9
d) ['5', '1', '7', '2', '4', '9']
e) ['5.1', '7.2', '4.9']

6.

a) 12
b) 22
c) 4
d) 16
Listing 1: q7.py

```python
'''Computes the number of times 5 dice have two pairs.'''

import random

def roll_dice():
    'Returns the rolls of two dice'
    dice = []
    for i in range(5):
        dice += [ random.randint(1,6)]
    return dice

def two_pair(dice):
    'Return True if dice has two pair property. Returns False otherwise'

    '''Use a variable count to keep track of the number of times
dice values appear.
    * count[0]: not used and stays 0,
    * count[1]: number of times the die value 1 appears,
    * count[2]: number of times the die value 2 appears,
    etc.
    '''
    count = [0] * 7

    if len(set(dice)) == 3:
        for die in dice:
            count[die] += 1
        count = sorted(count)

        '''in sorted order, the last 3 values in count
        must be 1, 2, 2, (in that order) for the dice to have
        the two pair property.
        '''
        if count[4:] == [1, 2, 2]:
            return True
    return False

def main():
    two_pair_count = 0
    print '*** Dice simulation to count the number of two pairs that occur. **'
    for i in range(1000):
        if two_pair(roll_dice()):
            two_pair_count += 1
    print 'Two pairs appeared %d out of %d rolls of five dice.' % ( two_pair_count, 1000)
main()
```
### Version B

1.
   a) 15  
   b) 80  
   c) 15  
   d) 27  
   e) -80

2.
   a) def, return, print  
   b) 1, 12, 8, 9, 3, 4, 5, 6, 9, 10, 12, 13  
   c) The value of result is 29.  
   d) NONE  
   e) n

3.
   a) 2  
   b) 3  
   c) `set([9, 'pear', 'apple', 7])` # set elements can be in any order  
   d) `set([9, 7])` # set elements can be in any order  
   e) `set(['pear'])`

4.
   a) `[0, 1, 2, 3, 4, 5]`

---

A plot for Exam #2

Line B

```
<table>
<thead>
<tr>
<th>week</th>
<th>dollars($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
```

b)
5.
   a) float
   b) string
   c) 8.1
   d) ['4', '5', '5', '3', '8', '1']
   e) ['4.5', '5.3', '8.1']

6.
   a) 15
   b) 555
   c) 6
   d) 36
   e) 43

7.

Listing 2: q7.py

```python
''' Computes the number of times 5 dice have two pairs. '''
import random

def roll_dice():
    'Returns the rolls of two dice'
    dice = []
    for i in range(5):
        dice += [random.randint(1,6)]
    return dice

def two_pair(dice):
    'Return True if dice has two pair property. Returns False otherwise'
    ''' Use a variable count to keep track of the number of times
dice values appear.
    * count[0]: not used and stays 0,
    * count[1]: number of times the die value 1 appears,
    * count[2]: number of times the die value 2 appears,
etc.
    '''
    count = [0] * 7
    if len(set(dice)) == 3:
        for die in dice:
            count[die] += 1
        count = sorted(count)
        ''' in sorted order, the last 3 values in count
        must be 1, 2, 2, (in that order) for the dice to have
        the two pair property.
        '''
        if count[4:] == [1, 2, 2]:
```
return True

return False

def main():
    two_pair_count = 0
    print '** Dice simulation to count the number of two pairs that occur. **'
    for i in range(1000):
        if two_pair(roll_dice()):
            two_pair_count += 1
    print 'Two pairs appeared %d out of %d rolls of five dice.' % (two_pair_count, 1000)
main()