CSCE 110: Programming I

Exam #2 — Answer Key

April 13, 2012

Your final score is circled in black. There are three values on your exam. The first, second, and third numbers represent the number of points you received for questions 1 through 7, question 8, and bonus points for attending class on Tuesday, March 27th, respectively.

Below are the solutions for Versions Gig 'Em and Reveille 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 April 24, 2012. 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 8 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 Exam Solutions

2.1 Version Gig 'Em

1.

  a) def, print, return
  b) 1, 3, 8, 13, 8, 9, 10, 3, 4, 5, 6, 10, 11, 13, 14
  c) 39
  d) n, x
  e) a, b, m, n
  f) n, result
2.
   a) 3
   b) set([10])
   c) set(['orange', 10, 'apple', 5])
   d) set(['apple'])
   e) set([25, 20])

3.

![Plot for Question #3.](image)

Figure 1: Plot for Question #3.

4.
   a) 30
   b) yesyes
   c) 1
   d) 9
   e) 8

5.
   a) [15, 30, 6]
   b) [[5, 25], [10, 100], [2, 4]]
   c) [11, -10]

6. Given that a red face is face up, the probability that the other side is 2/3. This problem is logically equivalent to the Monty Hall problem. Essentially, once we know that the face of the card is red, then there are 2 red faces and 1 green face remaining. Hence, the probability 2/3.
7. 
   a) 9  
   b) 15  
   c) False  
   d) True  
   
   e) The parameter a_list is a list of lists. So, f1() sums the individual lists in a_list and then compares that sum to the value given in the parameter total.

8. 
   There are many possible solutions to this problem. We will discuss alternative solutions in class. I present one solution to the problem below. For the sample output shown in the exam, the count for the number of full houses with the same color is incorrect. The program below prints the correct full house counts.

   Listing 1: full-house.py

   ```python
   """ Computes the number of times a full house of the same color occurs. """
   import random
   
   def roll_dice():
       dice = []
       for i in range(5):
           dice += [random.randint(1, 6)]
       return dice
   
   def full_house_same_color(dice):
       """ Returns True if the dice represents a full house of the same color. """
       """ Returns False otherwise. """
       sorted_dice = sorted(dice)
       if sorted_dice in [[1,1,6,6,6], [1,1,6,6,6], [2,2,5,5,5], 
       [2,2,5,5,5], [3,3,4,4,4], [3,3,4,4,4]]:
           return True
       else:
           return False
   
   def main():
       num_dice_rolls = 10000
       print 'RESULTS:'
       full_house_same_color_count = 0
       for i in range(num_dice_rolls):
           if full_house_same_color(roll_dice()) == True:
               full_house_same_color_count += 1
           print 'Full house with same color appeared %d out of %d rolls.' % (full_house_same_color_count, num_dice_rolls)
   
   main()
   ```
2.2 Version Reveille

1.
   a) def, print, return
   b) 1, 3, 8, 13, 8, 9, 10, 3, 4, 5, 6, 10, 11, 13, 14
   c) 51
   d) n, x
   e) a, b, m, n
   f) n, result

2.
   a) 2
   b) set(['pear'])
   c) set(['pear', 'grape', 17])
   d) set([17])
   e) set([10, 15])

3.

Figure 2: Plot for Question #3.
4.
   a) 35
   b) 3333
   c) 3
   d) 15
   e) 10

5.
   a) [30, 15, 12]
   b) [[10, 100], [5, 25], [4, 16]]
   c) [14, -4]

6. Given that a red face is face up, the probability that the other side is 2/3. This problem is logically equivalent to the Monty Hall problem. Essentially, once we know that the face of the card is red, then there are 2 red faces and 1 green face remaining. Hence, the probability 2/3.

7.
   a) 3
   b) 15
   c) True
   d) False
   e) The parameter a_list is a list of lists. So, f1() sums the individual lists in a_list and then compares that sum to the value given in the parameter total.

8. There are many possible solutions to this problem. We will discuss alternative solutions in class. I present one solution to the problem below. For the sample output shown in the exam, the count for the number of full houses with the same color is incorrect. The program on the next page prints the correct full house counts.
Listing 2: full-house.py

```python
import random

def roll_dice():
    dice = []
    for i in range(5):
        dice += [random.randint(1, 6)]
    return dice

def full_house_same_color(dice):
    '''Returns True if the dice represents a full house of the same color.
    Returns False otherwise.
    '''
    sorted_dice = sorted(dice)
    if sorted_dice in [[1, 1, 6, 6, 6], [1, 1, 1, 6, 6], [2, 2, 5, 5, 5],
                        [2, 2, 2, 5, 5], [3, 3, 4, 4, 4], [3, 3, 3, 4, 4]]:
        return True
    else:
        return False

def main():
    num_dice_rolls = 10000
    print('RESULTS:
    full_house_same_color_count = 0
    for i in range(num_dice_rolls):
        if full_house_same_color(roll_dice()) == True:
            full_house_same_color_count += 1
        print('Full house with same color appeared %d out of %d rolls.' %
              (full_house_same_color_count, num_dice_rolls))
main()
```