CS 140 Final Exam Review Problems

This is a **cumulative** final exam, so please review all of the practice problems as well as your quizzes and exam. There is some material that you haven't been tested on yet (images, strings, and lists), so most of the following practice problems are focused on those topics.

1. Write a function **`increase_red`** that takes an image as an argument and increases the red in the image by 10%. Write a complete Python program that class **`increase_red`** and displays the resulting image. Use any image in the gallery as an example.

```
def increase_red(surface):
    for y in range(surface.get_height()):
        for x in range(surface.get_width()):
            color = surface.get_at((x,y))

            # multiplying the red by 1.1 is the same
            # as increasing by 10%
            red = int(1.1 * color.r)

            # the above is a short cut, here's a potentially
            # more intuitive way to do this
            # tenpercent = int(color.r*0.1) # calculate 10% increase (force to be int)
            # red = color.r + tenpercent

            # Clip the red value at 255.
            if red > 255:
              red = 255
            surface.set_at((x,y),(red,color.g,color.b))

#--------------------------------------
# M a i n   P r o g r a m
#--------------------------------------
import pygame
pygame.init()
#pic = pygame.image.load("/home/stlawu/jchapman/t/Chapman/CS140/gallery/sunset.jpg")
# Here's how I do this from my windows machine
pic = pygame.image.load("T:/Chapman/CS140/gallery/sunset.jpg")
window = pygame.display.set_mode((pic.get_width(),pic.get_height()))
window.blit(pic,(0,0))
increase_red(window)
pygame.display.update()
raw_input("hit enter to continue")
pygame.quit()
```

CS 140 Final Exam Review Problems

2. Write a Python program that opens a pygame window and draws a random colored circle of radius 10 wherever the user clicks the mouse button (the click should be the center of the circle). The name of the event for a mouse click is MOUSEBUTTONDOWN. The program should also quit when the user clicks the X in the upper left of the pygame window.

```
import pygame
import random

# standard pygame stuff
pygame.init()
win = pygame.display.set_mode((300,300))

quit = False
white = (255,255,255)
win.fill(white)
pygame.display.update()

# keep running until the user
# kills the window
while not quit:

    # get the user's events
    for event in pygame.event.get():

        if event.type == pygame.QUIT:
            quit = True

        # check for a mouse click
        if event.type == pygame.MOUSEBUTTONDOWN:

           # get the position of the mouse click
           (x,y) = event.pos

           # generate a random color
           r = random.randrange(256)
           g = random.randrange(256)
           b = random.randrange(256)

           # draw a circle
           pygame.draw.circle(win, (r,g,b), (x,y),  10)
           pygame.display.update()

pygame.quit()
```

3. List practice
    a. What are the two ways to add something to a list? How are they different?

       listname.append(item) – this method puts this item at the end of the list
       listname.insert(location, item) – this method puts this item at the "location" spot in the list

    b. What are the two ways to remove something from a list? How are they different?

       listname.pop(location) – this method removes the item at the "location" spot AND returns the item that was in this spot; if no location is specified, the default is the last item
       listname.remove(item) – this method removes the FIRST occurrence of item in the list

c. What is the difference between a list and a tuple?

A list is *mutable* which means you can change any of the entries in the list; a tuple is *immutable* (so you can't change any of the entries in a tuple).

d. In the Python shell, do the following:

i. Define a variable named **`states`** that is an empty list

```
states = []
```

ii. Add **`Iowa`** to the list

```
states.append("Iowa")
```

iii. Now add **`Nebraska`** to the end of the list

```
states.append("Nebraska")
```

iv. Define a variable **`states2`** that is initialized with **`New York`**, **`Vermont`**, and **`New Hampshire`**

```
states2 = ["New York", "Vermont", "New Hampshire"]
```

v. Add **`Maine`** to the beginning of the list

```
states2.insert(0,"Maine")
```

vi. Add **`Massachusetts`** so that it is the third state in the list

```
states2.insert(2,"Massachusetts")
```

vii. Add **`Pennsylvania`** to the list so that it appears before **`New York`**. Do this as if you DO NOT KNOW where **`New York`** is in the list

```
# This is a new (to you) list operation

ind = states2.index("New York")
states2.insert(ind,"Pennsylvania")

# You could have written your own ".index" as follows:
ind = -1 # says NY is not in the list
for i in range(len(states2)):
    if states2[i] == "New York":
        ind = i
states2.insert(ind,"Pennsylvania")
```

viii. Remove the 5th state from the list and print that state's name

```
state5 = states2.pop(4)
print state5
# The 5th state should be Vermont
```

CS 140 Final Exam Review Problems

4. Write a short Python code segment that adds up the lengths of all the words in a list and then prints the average (mean) length. Use the final list from Problem 3 to test your program.

```
sum = 0
for i in states2:
    sum = sum + len(i)
print float(sum)/len(states2)
# note that the spaces in NY and NH count towards the length (answer = 10.2)

# another way
sum = 0
for i in range(len(states2)):
    sum = sum + len(states2[i])
print float(sum)/len(states2)
```

5. Write a short Python code segment that prints the longest word in a list. Again, use your final list from Problem 3 to test your program.

```
#5
longest = ''
for i in states2:
    if len(i) > len(longest):
        longest = i
print longest

# OR
longest = ''
for i in range(len(states2)):
    if len(states2[i])>len(longest):
        longest = states2[i]
print longest
```

Answer will be Massachusetts; really Massachusetts and New Hampshire have the same length (because spaces count), but because Massachusetts comes first in the list, it is named the longest (because that word is "seen" first by the program and when the program gets to New Hampshire, the length of NH doesn't exceed the length of MA).

6. Write a program that creates a list of all the integers less than 100 that are multiples of 3 or 5.

```
nums = []
for i in range(100):
    if i % 3 == 0 or i % 5 == 0:
        nums.append(i)

print nums
```

CS 140 Final Exam Review Problems

7. Define two variables **first** and **second** so that **first = "Jim"** and **second = "Dwight"**. Write a short Python code segment that swaps the values assigned to these two variables and prints the results.

```
first = "Jim"
second = "Dwight"

temp = first
first = second
second = temp

print first
      prints Dwight
print second
      prints Jim
```

8. Errors
   a. Does the program below have an error? If so, why?

```
t = (1, 'a', 9.2)
t[0] = 6
```

Yes. You cannot change the value of an element in a tuple (what the second line tries to do) because tuples are immutable.

   b. Does the program below have an error? If so, why?

```
t = [1, 'a', 9.2]
t[0] = 6
```

No. You can change the value of an element in a list (what the second line of code is doing) because lists are mutable.

   c. Does the program below have an error? If so, why?

```
t = [1, 'a', 9.2]
t[4] = 6
```

Yes. Will have an "index out of range" error because there is not an element in the 4 location of the list (would need to have at least 5 items in the list to have something in the 4 location).

   d. Does the program below have an error? If so, why?

```
t = 'hello'
t[0] = 'H'
```

Yes. Strings, like tuples, are immutable, so you can change an individual element in a string.

e. Assuming words is a valid list of words, the program below tries to print the list in reverse. Does it have an error? If so, why? (Hint: there are two problems with the code.)

```
for i in range(len(words), 0, -1):
    print words[i],
```

It should be:

```
for i in range(len(words)-1, -1, -1):
    print words[i],
```

The first entry in range is the starting point. Since the indexing in strings and lists starts at 0, the index of the entry in the last spot of the list is len(words)-1.

The second entry in range is one value beyond the end point. If you want to print all of the items in the list, including the very first one, the last index you need to use is 0. Since we are counting backwards, one value beyond 0 is -1.

9. Again, make sure you review the material that was covered on the midterm exam!