CS 2316
Exam 2 Spring 2012

Name : _____

Grading TA: _____

- INTEGRITY: By taking this exam, you pledge that this is your work and you have neither given nor received inappropriate help during the taking of this exam in compliance with the Academic Honor Code of Georgia Tech. Do NOT sign nor take this exam if you do not agree with the honor code.
- DEVICES: If your cell phone, pager, PDA, beeper, iPod, or similar item goes off during the exam, you will lose 10 points on this exam. Turn all such devices off and put them away now. You cannot have them on your desk.
- ACADEMIC MISCONDUCT: Academic misconduct will not be tolerated. You are to uphold the honor and integrity bestowed upon you by the Georgia Institute of Technology.
  - Keep your eyes on your own paper.
  - Do your best to prevent anyone else from seeing your work.
  - Do NOT communicate with anyone other than a proctor for ANY reason in ANY language in ANY manner.
  - Do NOT share ANYTHING during the exam. (This includes no sharing of pencils, paper, erasers).
  - Follow directions given by the proctor(s).
  - Stop all writing when told to stop. Failure to stop writing on this exam when told to do so is academic misconduct.
  - Do not use notes, books, calculators, etc during the exam.
- TIME: Don't get bogged down by any one question. If you get stuck, move on to the next problem and come back once you have completed all of the other problems. This exam has 5 questions on 10 pages including the title page. Please check to make sure all pages are included. You will have 50 minutes to complete this exam.

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community. I have also read and understand the requirements outlined above.

Signature: _____

Question	Points	Score
1. Multiple Choice	2	
2. Rank Days	10	
3. GUI drawing	13	
4. newsVender GUI	18	
5. Cheetah	5	
Total:	48	

1. *(2 points)*
   For each of the following multiple choice questions, indicate the most correct answer by circling it!

   (a) [1 pt] What is the purpose of writing a class to contain GUI code?

   A. It makes your code run faster.

   B. It allows us to share references to our widgets easily between methods by making the widgets instance variables.

   C. Widgets become globally accessible, no matter which function they are declared in.

   D. It allows Python to delete all widgets easily when the window is closed.

   E. It allows us to take advantage of polymorphism.

   (b) [1 pt] When reading in CSV files using the CSV Reader module:

   A. you do not have to manually close the file after you are finished.

   B. you can use readlines to retrieve all of the rows from the file as a list of lists.

   C. data is returned as a tuple of strings, where each string is one element in a row.

   D. you need to iterate through the reader object to retrieve each row.

2. *(10 points)*

Write a function named `rankDays` which takes in one parameter, aList. The list will be a list of tuples, with each tuple containing information in the format (Number Processed, Day of Week). The Day of Week will be a string (either "Monday", "Tuesday", "Wednesday", "Thursday", or "Friday"), and the Number Processed will be an integer. You should create and return a dictionary which has the day's "rank" (a number 1-5, with 1 corresponding to the day with the largest number of items processed) as a key and the corresponding day of the week as the value. The ranking criteria will be on the numbers of items processed each day. You may assume that no two days will share the same number of items processed (i.e. each day will have a uniquely different number of items processed).

Example test case:

```
>>> aList = [(456, "Monday"), (371, "Tuesday"), (2316, "Wednesday"),
(1371, "Thursday"), (17, "Friday")]
>>> myDict = rankDays(aList)
>>> print(myDict)
{1: 'Wednesday', 2: 'Thursday', 3: 'Monday', 4: 'Tuesday', 5: 'Friday'}
```

Solution:

```
def rankDays(aList):
   myDict = {}
   aList.sort(reverse = True)
   rank = 1
   for item in aList:
      myDict[rank] = item[1]
      rank = rank+1
   return myDict
```

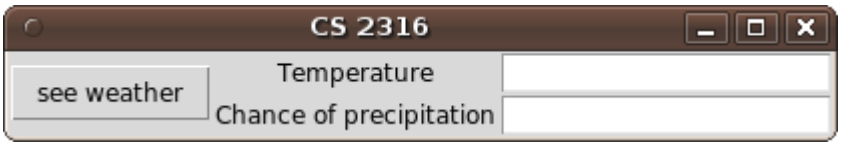
Grading: +1 correct function header
+1 uses dictionary to store values
+2 the keys in the dictionary are the numbers 1..5
+3 correctly ranks days based upon number of items processed.
(-1 if sort order is backwards)
+2 the value of each key is the correct day in the week.
+1 returns a dictionary
-2 if assigning none to a critical variable (e.g. alist=alist.sort())

3. *(13 points)*
Given the following code, draw the GUI that is produced TWICE. Draw it once as it first appears. Below that, draw the GUI a second time after the button is pressed. Include the window with any decorations. Indicate colors, shading, or state with arrows and labels.

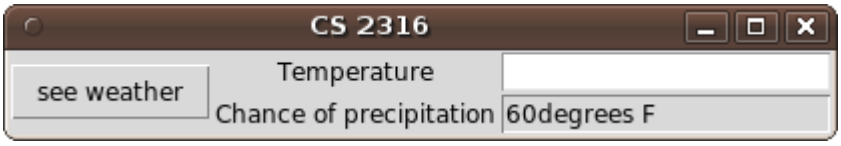
```
from tkinter import *
class MyScreen:
    def __init__(self,Window):
        self.var=StringVar()
        button1 = Button(Window, text="see weather", command=self.clicked)
        frame = Frame(Window)
        frame.pack(side=RIGHT)
        label1 = Label(frame, text="Temperature")
        label2  = Label(frame, text="Chance of precipitation")
        label1.grid(row=1,column=2)
        label2.grid(row=2, column=2)
        self.entry1 = Entry(frame, text="50degrees F")
        self.entry2 = Entry(frame)
        button1.pack(side=RIGHT)
        button1.config(relief=RAISED)
        self.entry2.grid(row=2, column=4, columnspan=3)
        self.entry1.grid(row=1, column=4)

    def clicked(self):
        self.entry2.config(state="readonly")
        self.entry2.config(textvariable=self.var)
        temp=self.entry2.get()
        self.var.set("60degrees F")

Window = Tk()
Window.title("CS 2316")
app = MyScreen(Window)
Window.mainloop()
```

Solution:

As appears:



After Button Press:



Grading:

```
+1 for the window with correct title!
+1 for "see weather" button existance
+1 for button on far left!

+1 for Temperature Label
+1 for Chance of precipitation label
+1 for correct location of Temp Label.
+1 for correct location of Perc. label.

+1 for top entry
+1 for bottom entry
+1 if both entries are in correct location.
+1 if BOTH entries are blank!

After button press:
 +1 if bottem entry is shaded or otherwise indicated as "read only"
 +1 if bottom entry has the text "60degrees F"
  (-1 for any other changes...)
```

This page intentionally left blank as drawing area for the GUI drawing question.

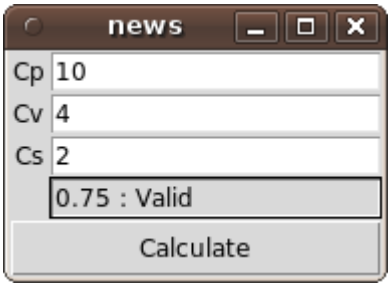
4. *(18 points)*

Write a program to creat a Graphical User Interface that looks like the picture below. Your program will calculate the critical fractile when the user presses the "Calculate" button, using this formula:

$(Cp - Cv)/(Cp - Cs)$

Display the result of your calculation in the bottom entry box. If any of the entries do not contain valid numbers, or if the result is larger than 1.0, or causes a divide by zero error, display "Invalid" in the fourth text entry. If all of the numbers are valid and the result is 1.0 or less, display the number in the fourth text entry field along with a colon (:) and the word "Valid" as displayed in the picture below.

Note that the title of the window is "news", the first three entries are normal, the forth entry is "read only", and the "Calculate" button spans the entire window.



Solution:

```
from tkinter import *
class NewsVendor():
    def __init__(self, rootWin):
        rootWin.title("news")
        Label(rootWin, text='Cp').grid(row=0,column=0)
        Label(rootWin, text='Cv').grid(row=1,column=0)
        Label(rootWin, text='Cs').grid(row=2,column=0)
        self.cp=StringVar()
        self.cv=StringVar()
        self.cs=StringVar()
        self.res=StringVar()
        Entry(rootWin, textvariable=self.cp).grid(row=0, column=1)
        Entry(rootWin, textvariable=self.cv).grid(row=1, column=1)
        Entry(rootWin, textvariable=self.cs).grid(row=2, column=1)
        e1 = Entry(rootWin, textvariable=self.res,state='readonly')
        e1.grid(row=3, column=1 )
        b1 = Button(rootWin, text='Calculate',command=self.clicked)
        b1.grid(row=4, column=0, columnspan=2, sticky=EW)
```

```
    def clicked(self):
        try:
            top = float(self.cp.get()) - float(self.cv.get())
            bottom=float(self.cp.get()) - float(self.cs.get())
            opt =  top / bottom

            if opt<=1:
                self.res.set(str(opt)+' : '+'Valid')
            else:
                self.res.set('Invalid')
        except:
            self.res.set('Invalid')


rootWin=Tk()
app = NewsVendor(rootWin)
rootWin.mainloop()
```

Grading:
+1 point for importing tkinter
+2 for creating GUI (Tk() call and Object creation)
+1 point for correct title on window
+3 points for correct labels and position.
+3 points for correct entries and position (forth entry is read-only!)
+2 points for Calculate button (which spans the window).
+1 point if Calculate button calls a function that tries to do math.
+2 points for correct math.
+2 points for correctly updating the "results" entry, leaving it read-only.
+1 point for correctly display Invalid on exceptions (number conversion or divide by zero)

This page intentionally left blank for solution to the newsVender problem.

5. *(5 points)*
Examine the following code

```
class Cheetah:
    def __init__(self,speed):
        self.speed=speed
        self.coat='white'
    def printspeed(self):
        print("My cheetah runs at a speed of",self.speed,"mph.")
    def changecolor(self,color):
        self.printspeed()
        self.coat=color
        color=''
        return print("My cheetah is "+self.coat)

mycheetah=Cheetah(70)
print(mycheetah.coat)
print(mycheetah.changecolor('yellow'))
```

What is printed on the screen as the code above is executed?

Solution: white
My cheetah runs at a speed of 70 mph.
My cheetah is yellow
None

Grading:
+1 for each correct answer. +1 bonus if all 4 answers are correct