CS 1803 Spring 2011 Exam 2

Name: _____ Section:_____
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.
- 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. 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.

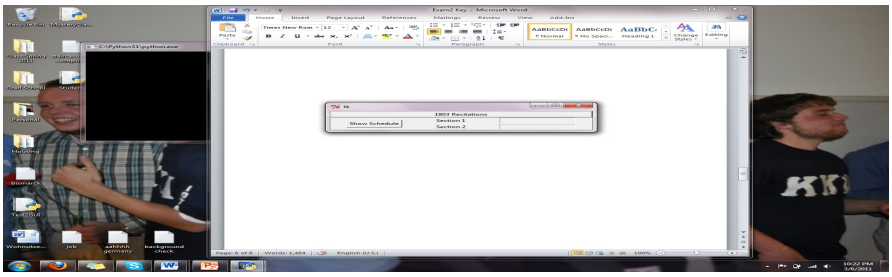
Signature: _____

Question	Possible Points	Earned Points	Graded By
GUI	20		
Code Reading	7		
Staircase	15		
Multiple Choice	14		
Total Points	57		

Question 1: GUI Reading (20 points)

```
from tkinter import *
class Test2GUI:
    def __init__(self,root,classSections):
        self.classSections = classSections
        frame = Frame(root)
        frame.pack()
        label = Label(frame, text = "1803 Recitations", width = 60)
        button = Button(frame, text = "Show Schedule",
                                command = self.clicked)
        section1 = Label(frame, text = "Section 1")
        section2 = Label(frame, text = "Section 2")
        self.entry1 = Entry(frame, state = "readonly")
        self.entry2 = Entry(frame, state = "readonly")
        label.grid(row = 0, column = 0, columnspan = 3)
        label.config(relief = RAISED)
        button.grid(row = 1, column = 0, rowspan = 4)
        section1.grid(row = 1, column = 1)
        section2.grid(row = 2, column = 1)
        self.entry1.grid(row = 1, column = 2)
        self.entry2.grid(row = 2, column = 2)
    def clicked(self):
      #Write the code that allows the text in both entry widget to be changed

      self.entry1.config(state = "normal")                +2 (1 for each line)
      self.entry2.config(state = "normal")


      #write the code to remove the existing text and then insert the values from the classSections list
      #into the 2 entries (the first element in the list should go in the first entry widget and the second
      #element in the list should go in the second entry widget) – You must update both entry
      #widgets
      self.entry1.delete(0,END)                           +1 for deleting
      self.entry2.delete(0,END)
      self.entry1.insert(0,self.classSections[0])         +1 for inserting correctly
      self.entry2.insert(0,self.classSections[1])         +2 for correct list indexing


      #write the code to set the entries back to read only


      self.entry1.config(state = "readonly")                  +2 (1 for each line)
      self.entry2.config(state = "readonly")


times = ["6:00-7:30","4:30-6:00"]
rootwin = Tk()
mywin = Test2GUI(rootwin, times)
rootwin.mainloop()
```

Draw the GUI that is created by running the above code, **before** the button is pushed.



+2 for each widget- 1 for correct widget, 1 for correct placement (12 points total)

- 1803 Recitations Label
- Show Schedule Button
- Section1 Label
- Section2 Label
- Section1 Entry
- Section2 Entry

Question 2: Code Reading (7 Points)

Assume the following code is run:

```
class Student:
      def __init__(myself,name,age):
        myself.name = name
        myself.age = age
      def printSelf(myself):
        print("My name is:",myself.name)
        print("My age is:",myself.age)
      def sleep(myself):
        print("Trying to sleep...")
        print("Oh no! I forgot my name!")
        name = ""
        print("Cannot sleep at Georgia Tech.")

jason = Student("Jason",21)
jason.sleep()
jason.printSelf()
```

Write exactly what is printed to the screen when this code is run:

Solution/Grading:

Trying to sleep...	+3 for printing the 3 lines from the call to sleep()
Oh no! I forgot my name!	
Cannot sleep at Georgia Tech.	
My name is: Jason	+3 printing full line with correct name (+1 if incorrect name)
My age is: 21	+1 correct age

Question 3 – Code Writing (15 Points)

Write a function named generateStaircase that takes in two parameters, the number of rows and a filename. The function will generate a staircase of numbers and write this staircase out to a csv file (if you use python modules, remember to import them!). The staircase will be in the format as follows:
the first row will have a single 1 in it, the second row will have two 2's in it, the third row will have three 3's in it. For example,
generateStaircase(5, "test.csv")
will make a CSV file with the following format:
1
2,2
3,3,3
4,4,4,4
5,5,5,5,5

Solution/Grading:

```
def staircase(num, filename):          +2 correct method header
    import csv
    file = open(filename, "w")         +3 opens the correct file (+1 for a hard coded file)
    writer = csv.writer(file)
    for row in range(1,num):
        temp = []
        for col in range(row):
            temp.append(row)           +4 creates the correct row to add to the file
        writer.writerow(temp)          +4 writes to the csv file (+2 for forgetting to cast to string)
    file.close()                       +2 closes the file
```

-2 does not import correct module
-2 does not reset row

Question 4 - Multiple Choice (14 Points)

The following questions in this section may contain multiple correct answers. For those which do contain multiple correct answers, circle all correct answers. All questions will contain at least one correct answer. For each correct answer choice you select, you will receive one point. For each incorrect answer you select, you will lose half a point. You cannot earn below zero points on any one question.

Write your answers in the answer blanks provided above **IN ALPHABETICAL ORDER**.

1. ACE_____ 4. BCD_____

2. D_____ 5. BC_____

3. BD_____ 6.BCD _____

1. Which of the following statements is/are true of dictionaries?

    a. They may have any type as a value
    b. They are an immutable data type
    c. The dictionary's keys must all be immutable types
    d. The dictionary's values must all be immutable types
    e. Tuples may be used as keys

2. Which of the following are true about a **Subclass**?
    a. The class header must contain the keyword "extends"
    b. If a call to the superclass is used it must be the first thing in the method
    c. You must redefine all of the methods from the parent class
    d. It is acceptable to have a method with the same name and parameters as one in the parent class

3. Which of the following are mutable data types?
    a. Tuples
    b. Lists
    c. Strings
    d. Dictionaries

4. Assume you're coding a GUI. You have the following lines of code:

```
self.e1 = Entry(rootwin, width = 60).pack()
self.e2 = Entry(rootwin, width = 60).pack()
self.e3 = Entry(rootwin, width = 60)
self.e3.pack()
```

Which of the following statements is (are) correct?

    a. We can get and set the text in the entry box self.e1
    b. We can get and set the text in the entry box self.e3
    c. The variables self.e1 and self.e2 point to None.
    d. All three entry boxes will be displayed in the GUI.

For questions 5 & 6, consider the following code:

```
class MysteryClass():
      def __init__(self, a, b, c):
            self.a = a
            self.b = b
            self.c = c

      def mystery1(self, x, y):
            a = self.a + x
            b = self.b + y
            return (a, b)
      def mystery2(self):
            print(self.a + self.b + self.c)

class MysteryClass2(MysteryClass):
      def __init__(self, y, z):
            super().__init__(y, z, None)
            self.c  = "CS 1803"
            self.d = "Exam 2"
      def mystery1(self, a):
            return (self.d, self.c)
      def mystery2(self, y):
            super().mystery2()
            return (self.d, y)

m2 = MysteryClass2("This", "is")
print(m2.mystery2("test"))
print(m2.mystery1("math 2602"))
```

5. What happens when the above code is run?
    a. An error occurs
    b. Three lines are printed
    c. ThisisCS 1803 is printed
    d. ('Exam 2', None) is printed
    e. ThisisNone is printed

6. What happens when the above code is run?
    a. MysteryClass's mystery1 method is run.
    b. MysteryClass's mystery2 method is run.
    c. MysteryClass2's mystery1 method is run.
    d. MysteryClass2's mystery2 method is run.