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 will result in a substantial grade penalty.
  - 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 4 questions on 8 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	12	
2. Exam Grades	9	
3. GUI drawing	8	
4. count T's	9	
Total:	38	

1. *(12 points)*
For each of the following multiple choice questions, indicate the single most correct answer by circling it!

(a) [1 pt] Consider a table named ORDERS with columns CUSTOMER and TOTAL representing the customers name and the order total. (A single customer may have more than one order.) Which of the following SQL statements will generate a report of all customers who have at least one order larger than the average order total? (Without duplicate names in the report...)

A. `SELECT DISTINCT CUSTOMER FROM ORDERS WHERE TOTAL > AVG(TOTAL)`

B. `SELECT CUSTOMER FROM ORDERS WHERE TOTAL > AVG(TOTAL) DISTINCT`

C. `SELECT DISTINCT * FROM ORDERS WHERE TOTAL > AVG(TOTAL)`

D. `SELECT DISTINCT CUSTOMER FROM ORDERS WHERE TOTAL >`
`(SELECT AVG(TOTAL) FROM ORDERS)`

(b) [1 pt] Given the following regular expression, choose the answer that it would fully match: `[(abc){1,2}]`

A. `(abc){1,2}` B. `abcabc` C. `ab` **D.** `2`

(c) [1 pt] Given the following regular expression, choose the answer that it would match: `johndoe-[^A-Za-z]+\.(?:com|org|net)`

A. `johndoe-6@1234.org`

B. `johndoe-D@1234.net`

C. `johndoe-768@smart.net`

D. `johndoe-gmail.com`

(d) [1 pt] An error in a program that makes it impossible to parse – and therefore impossible to interpret, is a:

A. Syntax Error B. Semantic Error C. Runtime Error D. Parse Error

(e) [1 pt] An error (in code) that leads to unexpected behavior. The program functions correctly (does what the code says) but the code does not actually perform the action that the programmer intended, is a:

A. Syntax Error **B. Semantic Error** C. Runtime Error D. Unexpected Error

(f) [1 pt] An error raised by the python interpreter while the program is executing if something goes wrong. For example, a divide by zero error, is a:

A. Syntax Error B. Semantic Error **C. Runtime Error** D. Interpreter Error

(g) [1 pt] Which of the following is/are valid functions to use with writing to a CSV file using a CSV Writer?

A. write
B. writeline
C. writerow
D. writelines
E. writerows
F. A and C are valid choices.
G. B and D are valid choices.
H. C and E are valid choices.

(h) [1 pt] Which of the following function calls (using pymysql) would result in an exception?

A. cursor.commit()
B. db.commit()
C. db.close()
D. cursor.close()
E. cursor.fetchone()

(i) [1 pt] What is the name of the option used in the constructor to associate a radiobutton with a Tkinter variable in order to keep track of which radiobutton is pressed?

A. `value` B. `StringVar` **C. `variable`** D. None of these

(j) [1 pt] When an entry box has state set to DISABLED, the user cannot alter the text it contains, but the program can by using the `insert` method.

A. True **B. False**

(k) [1 pt] What is pointed at by the `myList` and `sortedList` variables after the following code is executed?

```
myList = [128,100,111,31]
sortedList = myList.sort()
```

A. myList = [128,100,111,31] and sortedList = [31,100,111,128]
B. Both myList and sortedList = [31,100,111,128]
C. myList = None and sortedList = [31,100,111,128]
D. sortedList = None and myList = [31,100,111,128]
E. sortedList = [0,1,2,3] and myList = [128,100,111,31]
F. myList = [128,100,111,31] and an exception occurs, preventing sortedList from being assigned.

(l) [1 pt] Suppose you want to extract all dates from a string of text, myText. The date will always be in the format YYYY-MM-DD. Which of the following will return a list of only these date strings?

A. `theDates = findall("[0-9]{4}.*[0-9]{2}.*[0-9]{2}", myText)`

B. `theDates = findall("\D{4}-\D{2}-\D{2}", myText)`

C. `theDates = findall("\d*-\d*-\d*", myText)`

D. `theDates = findall("\d{4}-\d{2}-\d{2}", myText)`

E. `theDates = findall("\S{4}-\S{2}-\S{2}", myText)`

2. *(9 points)*
A table has been created for you with the following command:
CREATE TABLE 2316EXAMS (STUDENT TEXT NOT NULL, GRADE INTEGER, EXAMNO INTEGER)

The database has contents such as the following (but with many more records):

STUDENT	GRADE	EXAMNO
Michael Jordan	76	1
Reese Witherspoon	52	1
James Franco	10	1
Michael Jordan	77	2
Reese Witherspoon	65	2
James Franco	98	2
Michael Jordan	68	3
Reese Witherspoon	60	3
James Franco	50	3

Based on this table, write the SQL statements which will complete each task:

(a) [1 pt] Display the total class average:

Solution: `SELECT AVG(GRADE) FROM 2316EXAMS`

(b) [2 pts] Display Michael Jordan's Highest Exam Grade:

Solution:

```
SELECT MAX(GRADE) FROM 2316EXAMS WHERE STUDENT="Michael Jordan"
-or-
SELECT GRADE FROM 2316EXAMS WHERE STUDENT=’Michael Jordan’ ORDER BY
GRADE DESC LIMIT 1;
```

Grading: +1 for getting highest grade, +1 for selecting where student is Michael Jordan.

(c) [3 pts] Display each student name and their individual average (across all their exams) order the output from highest average to lowest:

Solution:

```
SELECT STUDENT, AVG(GRADE) FROM 2316EXAMS
    GROUP BY STUDENT ORDER BY AVG(GRADE) DESC!
```

Grading: +1 for selecting both student and avg(grade). +1 for grouping by student, +1 for ordering by avg(grade) DESC.

(d) [3 pts] Display the name of the student who did the best on exam 2:

Solution:

```
SELECT STUDENT FROM 2316EXAMS WHERE EXAMNO=2 ORDER BY GRADE DESC LIMIT 1
--or--
SELECT STUDENT FROM 2316EXAMS WHERE EXAMNO=2 AND GRADE =
    (SELECT MAX(GRADE) FROM 2316EXAMS WHERE EXAMNO=2)
```

Grading: +1 for SELECT STUDENT (only). +1 for WHERE EXAMNO=2. +1 for limiting to single maximum grade result.

3. *(8 points)*
The contents of the **emails.txt** file is four lines as follows:

```
johnsmith@gatech.edu
janedoegmail.com
123gojackets@yahoocom
hello
```

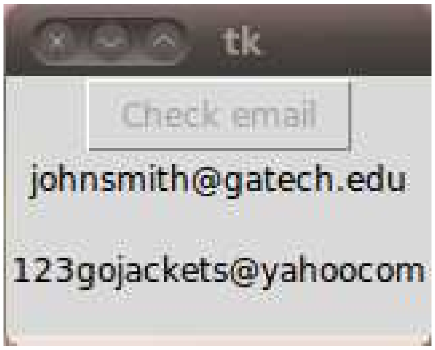
Given the following code draw a single picture that represents the GUI that is produced after the code is ran and the user has clicked the button 3 times. Include the window with any decorations. Indicate colors, shading, or state with arrows and labels.

```
from tkinter import *
import re

regexp = "[^.@]+@[a-z]+.(com)|(edu)"
f = open('emails.txt')
win = Tk()
b = Button(win, text="Check email")
b.pack()

def func():
    line = f.readline()
    if line == "":
        b.config(state=DISABLED)
        f.close()
    else:
        fa = re.findall(regexp, line)
        if len(fa) > 0:
            Label(win, text=line).pack()
        else:
            b.config(text="Invalid")
        line = f.readline()

b.config(command=func)
win.mainloop()
```

Solution: GUI: 

johnsmith@gatech.edu

123gojackets@yahoocom

Grading:
+1 window correctly drawn with decorations
+1 Button centered AND has ”Check Email” text.
+1 Button at top.
+1 Button state disabled (grayed out)
+1 johnsmith@gatech.edu text (anywhere)
+1 123gojackets@yahoocom (anywhere)
+2 (any) two lines of text in correct order (same as email.txt) under button

4. *(9 points)*
Bud Peterson is worried that somebody may be stealing T's. You are to write a function named countTs which will accept a string representing the URL of a website. Your objective is to download the HTML from this website and return an integer representing the number of times a letter ”T” (upper or lowercase) occurs.

Solution:

```
import urllib.request
from re import findall

def countTs(website):
    response = urllib.request.urlopen(website)
    html = response.read()
    text = html.decode()
    data = findall("[Tt]", text)
    return len(data)
// Or, if they didn't use regular expressions, they could do this:
    counter = 0
    text = text.lower()
    index = text.find("t")
    while index != -1:
       index = text.find("t",index+1)
       counter = counter+1
    return counter
```

Grading:
1 point for correctly importing urllib
3 points for correctly downloading the HTML.
4 points for counting the Ts (using re's or otherwise) (+3 if their code almost works, +2 if it would work with minor fixes, +1 if they have the right idea but the code is horribly wrong.)
1 point for returning an integer