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 6 questions on 13 pages including the title page. Please check to make sure all pages are included. You will have 1 hour and 45 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	20	
2. Short Answer	14	
3. Recursive Function	9	
4. Alternate Case	10	
5. GUI Calculator	28	
6. Get Items	10	
Total:	91	

1. *(20 points)*
For each of the following multiple choice questions, indicate any and all correct answers! Each question will have at least one correct answer. Indicate your selected answer(s) by circling it.

(a) Which of the following is/are true of GUIs in Python:

A. GUIs must be created inside of a class.

B. Pack and grid may not be used simultaneously in the same container.

C. We can change the text contained in an entry box by using the config method on the entry box.

D. When we create widgets inside of an init method, they are globally accessible.

(b) Which of the following is/are true about buttons in Tkinter?

A. Buttons can respond to keyboard events as well as mouse events.

B. `b=Button(rootWin, command=self.clicked() )` will properly call the self.clicked method when the button is clicked.

C. Buttons must be created as instance variables.

D. An error is thrown if a command is not specified for a button when the button is created.

(c) An instance variable:

A. is a variable which is accessible from any function in your Python file.

B. is always declared inside of an init method.

C. is contained within a class rather than an object.

D. is accessible from any method inside of the class.

E. belongs to a specific object, rather than the class.

(d) What is the purpose of writing a class to contain GUI code?

A. It allows us to create multiple copies of the GUI easily.

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.

(e) 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 list of strings, where each string is one element in a row.

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

(f) Which of the following is/are valid functios to use with writing to a CSV file using a CSV Writer?

A. write

B. writeline

C. writerow

D. writelines

E. writerows

(g) Indicate the string(s) below that will be matched by the regular expression: `\d{5}(?:-\d{4})?`

A. 1111-2222

B. 12345-2345

C. abcde-1234

D. 1234-abcde

E. 12345

F. 12345-1234

(h) Before scraping a webpage, it is necessary to:

A. Ensure that the website's Terms of Service permit web scraping.

B. Guarantee that the website does not have any form elements.

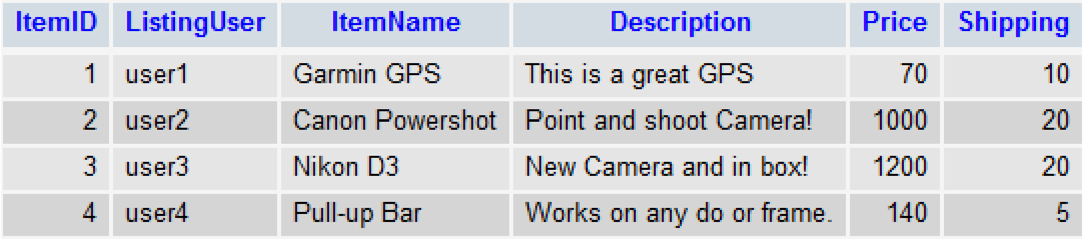
C. Ensure that the webpage does not contain any CSS (Cascading Style Sheets)

D. Import urllib

(i) Which of the following are aggregate functions in MySQL?

A. SUM B. ABS C. NOW D. ROUND

Use the MySQL table named Item to answer the following three questions.



ItemID	ListingUser	ItemName	Description	Price	Shipping
1	user1	Garmin GPS	This is a great GPS	70	10
2	user2	Canon Powershot	Point and shoot Camera!	1000	20
3	user3	Nikon D3	New Camera and in box!	1200	20
4	user4	Pull-up Bar	Works on any do or frame.	140	5

Note that the ItemID field is a unique, integer, auto-incrementin field, and that Price and Shipping are also integer fields. All other fields are Text fields.

(j) Which of the following queries will return the number of rows in the table?

A. `SELECT id FROM Item`

B. `SELECT SUM(id) FROM Item WHERE id > 0`

C. `SELECT COUNT(Shipping) FROM Item`

D. `SELECT COUNT(*) FROM Item`

(k) What is the function of the following SQL statement?

```
UPDATE Item SET shipping = 0 WHERE shipping <
                        ( SELECT AVG(shipping) FROM Item)
```

A. Returns all items that have a shipping amount of less than zero.

B. Sets all items to have a shipping amount of zero.

C. Sets items with a shipping amount that is less than the average to zero

D. Returns items with a shipping amount less than the average shipping amount.

(l) What would be the result returned by the following SQL query?

`SELECT COUNT(shipping) FROM Item`

A. 1 B. 2 **C. 3** D. 4 E. 0

2. *(14 points)*

For each of the following questions, give a brief answer:

(a) [3 pts] Write a regular expression that will match usernames, stored as strings, which adhere to the following conventions:

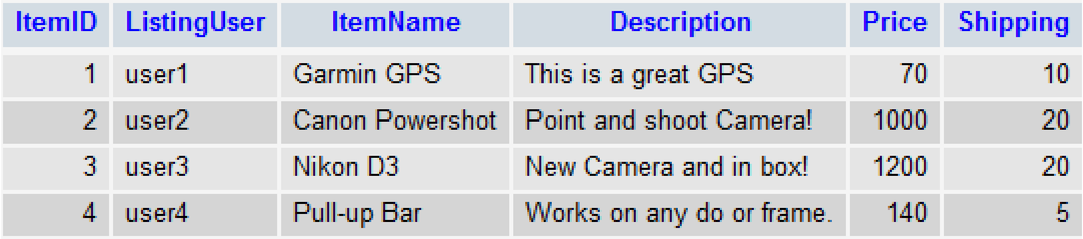
1. The username must start with three to ten letters, either upper or lowercase.
2. The username must end with at least one, but possibly more, digits.

Solution: Answer: (answers may vary)

[a-zA-Z]3,10+

Grading: +1 for A-Za-z. +1 for 3,10 +1 for +

Using the following SQL table named Item, write the SQL queries described below:



ItemID	ListingUser	ItemName	Description	Price	Shipping
1	user1	Garmin GPS	This is a great GPS	70	10
2	user2	Canon Powershot	Point and shoot Camera!	1000	20
3	user3	Nikon D3	New Camera and in box!	1200	20
4	user4	Pull-up Bar	Works on any do or frame.	140	5

(b) [2 pts] Write a query that will return all records that were posted by user1 and have shipping of 20 or greater.

Solution: SELECT * FROM Item WHERE ListingUser = ”user1” AND Shipping >= 20

(c) [2 pts] Write a query that will return the total cost of the item (price plus shipping) along with the item name of each item in the table.

Solution: ANSWER: SELECT Price + Shipping, ItemName FROM Item

(d) [3 pts] Insert the following information into the table as a new row. Do not specify an ItemID in your query.
ListingUser: user1
ItemName: Asus Notebook
Description: It works!
Price: 100
Shipping: 13

Solution: ANSWER: INSERT INTO Item (ListingUser, ItemName, Description, Price, Shipping) VALUES (”user1”, ”ASUS Netbook”, ”It works!”, 100, 13)

(e) [4 pts] Write a query that will return the highest priced item for each user along with the username, sorted from highest to lowest price.

Solution: ANSWER: SELECT ListingUser, MAX(Price) FROM Item GROUP BY ListingUser ORDER BY Price DESC

3. *(9 points)*
Pretend you are the Python interpreter and the following code has been entered and executed. Write what is printed to the screen beside the code:

```
def foo(num):
    print("start recursion!")
    print(num)
    if num <= 0:
        print("stop")
        return 0
    elif num == 1:
        print("close")
        return 1
    final = foo(num-1)*2
    print("test")
    return final

def funct(num):

    go = True
    while go:
        print("loop running")
        if num%3 == 0:
            go = False
            print("change go")
        num = num - 1
        print("fun")
    return num
    print("finished")

print( foo(3) )
print( funct(11) )
```

Solution:

```
start recursion!
3
start recursion!
2
start recursion!
1
close
```

```
test
test
4
loop running
fun
loop running
fun
loop running
change go
fun
8
```

Grading: +3 for the ”start, 3, start, 2, start, 1, close” sequence (+1 all three starts) (+1 all three numbers) (+1 for close)
+1 for each ”test”
+1 for the ”4”
+2 for ”loop running, fun, loop running, fun, loop running, fun” sequence (+1 all three loop runnings) (+1 for all three funs)
+1 for ”change go”
+1 for the ”8”

4. *(10 points)*
Write a function named `alternateCase` that takes in a string parameter and returns a string. The function takes the input string and changes each letter in it to alternate between lowercase and uppercase (starting with a lowercase). It then returns the modified string.

Example test cases:

```
>>> result =  alternateCase("CSisfun")
>>> print(result)
'cSiSfUn'
>>>
```

Solution:

```
def alternateChars(data):
    num = 0
    final = ""
    for letter in data:
        if num%2 == 0:
            final += letter.lower()
        else:
            final += letter.upper()
        num += 1
    return final
```

Grading:

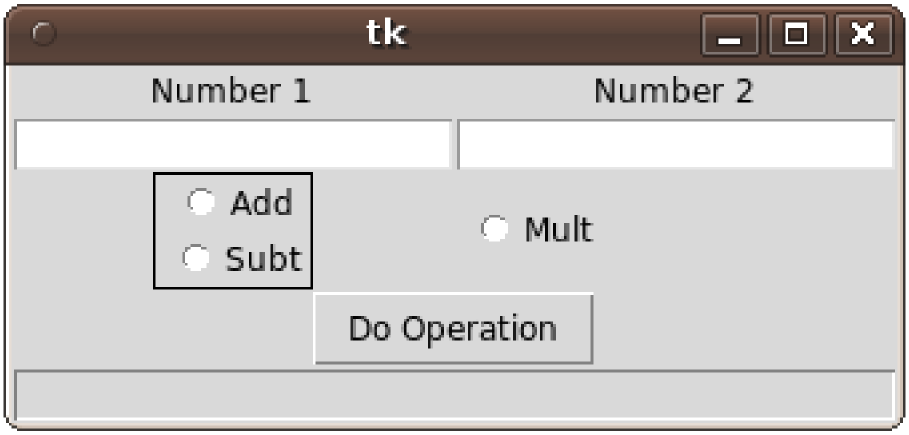
+1 correct method header
+1 looping through the string
+3 if statement (alternating letters)
+2 adding uppercase correctly (+1 correctly appending letter)
+2 adding lowercase correctly (+1 correctly appending letter)
+1 return a string

5. *(28 points)*
Pretend you are the python interpreter and the following code is executed. (NOTE: The Button is NOT pressed!)

```
from tkinter import *

class Calculator:

    def __init__(self, root):

        self.window = root
        label = Label(root, text = "Number 1")
        newLabel = Label(root)
        newLabel.config(text = "Number 2")
        label.grid(row = 0, column = 0)
        newLabel.grid(row = 0, column = 1)
        self.createEntries()
        frame = Frame(root, border=1, relief=SOLID)
        self.value = IntVar()
        add = Radiobutton(frame, text = "Add", variable = self.value, value = 1)
        add.pack()
        sub = Radiobutton(frame, text = "Subt", variable = self.value, value = 2)
        sub.pack()
        frame.grid(row = 2, column = 0)
        mult= Radiobutton(root, text = "Mult", variable = self.value, value = 3)
        mult.grid(row = 2, column = 1, sticky = W)
        b = Button(root, text = "Do Operation", command = self.operate)
        b.grid(row = 3, column = 0, columnspan = 2)
        self.answer = Entry(root, state= "READONLY")
        self.answer.grid(row=4, column = 0, columnspan = 2, sticky = W + E)

    def createEntries(self):
        self.first = Entry(self.window, state = NORMAL)
        self.first.grid(row = 1, column = 0)
        self.second = Entry(self.window, state = NORMAL)
        self.second.grid(row = 1, column = 1)

    def operate(self):
        pass

win = Tk()
calc = Calculator(win)
win.mainloop()
```

Draw the GUI that the code above creates! Make sure to include any and all window decorations.

Solution:



Grading:
+1 window outline
+1 correct title
+2 two labels on the first row
+3 two entries on the second row (+2 existance, +1 normal)
+1 frame depiction (solid border around Add/Subt buttons)
+2 right radio buttons in the center (+1 for existance)
+2 left radio button stickied left (+1 for existance)
+2 button is in the center of row 3
+2 answer entry is "readonly" and extends entire row

Now, re-write the operate method so that it takes the values in the self.first and self.second entries and either adds, subtracts, or multiplies them together depending upon the radio button that is selected. The final answer should be displayed in the self.answer entry. You may assume that the text in the self.first and self.second entries are valid floats and that a single radio button is selected.

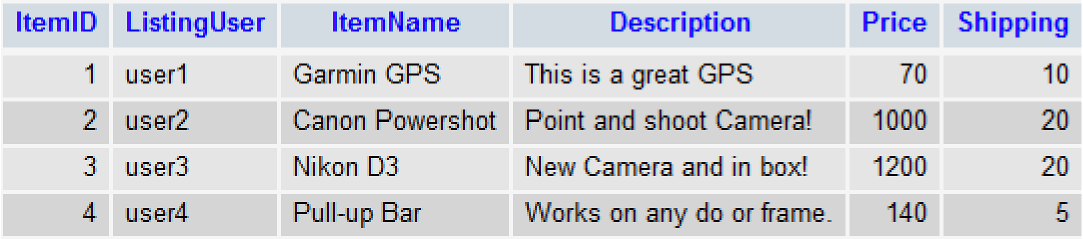
Solution:

```
def operate(self):
   num1 = self.first.get()
   num1 = float(num1)
   num2 = self.second.get()
   num2 = float(num2)
```

```
        if self.value.get() == 1:
            final = num1 + num2
        elif self.value.get() == 2:
            final =  num1 - num2
        elif self.value.get() == 3:
            final = num1 * num2
        self.answer.config(state=NORMAL)
        self.answer.delete(0,END)
        self.answer.insert(0, str(final) )
        self.answer.config(state="readonly")
```

Grading:
+3 retrieve(+2) and float(+1) both numbers.
+3 check for each value of the radio button intvar
+3 does correct operation
+3 does update correctly +2 for handling state, +1 for delete/insert

6. *(10 points)*



ItemID	ListingUser	ItemName	Description	Price	Shipping
1	user1	Garmin GPS	This is a great GPS	70	10
2	user2	Canon Powershot	Point and shoot Camera!	1000	20
3	user3	Nikon D3	New Camera and in box!	1200	20
4	user4	Pull-up Bar	Works on any do or frame.	140	5

Assume that your database has the above SQL table named Items. Write a function named `getItems` that accepts four parameters (username, hostname, password, and databasename). Your function should connect to the MySQL database specified by the parameters. Then it should query the above Items table, retrieve all of the unique item names in the database, and return a list of those items names.

Solution:

```
import pymysql

def getItems(uname, hostname, pwd, dbname):
itemslist = []

db = pymysql.connect(user = uname, passwd = pwd, host = hostname, db = dbname)
c = db.cursor()

sql = "SELECT DISTINCT ItemName FROM Item"
c.execute(sql)

for row in c:
itemslist.append(row[0])
                c.close()
                db.close()

return itemslist

+1: Importing pymysql
+1: Correct header
+2: Correct connection to database
+1: Using pymysql.connect
+1: Providing all connection parameters
+1: Correct SQL String
+1: Gets cursor and executes query on cursor
+2: Gets item names out of cursor and appends to list
+1: closes at least one of cursor or db, preferably both!
```

+1: Returns correct list of item names.