

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	8	
2. Multiple Choice	4	
3. Heat	9	
4. GUI coding	16	
5. count T's	9	
Total:	46	

1. (8 points)

For each of the following multiple choice questions, indicate the single most correct answer by circling it!

(a) [1 pt] What is the correct syntax for adding a number (1) to a tuple?

- A. `myTuple = myTuple + (1,)`
- B. `myTuple.append(1)`
- C. `myTuple.insert(0,1)`
- D. `myTuple = myTuple + 1`

(b) [1 pt] What will the '_' wildcard character in SQL match?

- A. **Exactly one character**
- B. One or more characters
- C. Any number of characters (including zero)
- D. '_' is not a proper SQL wildcard

(c) [1 pt] What is the type of the variable "var" after the following line of code is executed?

```
var=Label(win, text="Hello World.").pack()
```

- A. `'tkinter.Label'`
- B. `<class 'tkinter'>`
- C. `<class 'tkinter.Label'>`
- D. `<class 'NoneType'>`
- E. None of the above.

(d) [1 pt] Which of the following statements about Radiobuttons are false?

- A. Radiobuttons have a 'state' parameter
- B. Radiobuttons have a 'variable' parameter
- C. Radiobuttons have a 'value' parameter
- D. Radiobuttons can change the value of StringVar's
- E. **All of the above statements are true.**

- (e) [1 pt] Assume the class `Test1` creates a GUI and has been defined correctly. Examine the following lines of code:

```
rootWin = Tk()
question = Test1(rootWin)
```

Which of the following lines of code, if executed after the above code, will cause Python to throw an error?

- A. `Test1.difficulty = "moderate"`
 - B. `experience = Test1(rootWin)`
 - C. `rootWin.mainloop()`
 - D. `app.Label(rootWin, text = "LeBron")`
 - E. `question.answer = "here?"`
- (f) [1 pt] Given the following code, where will the label be placed in the `rootWin`?
- ```
from tkinter import*
rootWin = Tk()
l=Label(rootWin,text="Pumpkin Pie")
l.grid(row=10,column=31)
rootWin.mainloop()
```
- A. **The top left corner**
  - B. The bottom right corner
  - C. Somewhere in the middle according to the specified row/column numbers
  - D. It will not be shown in the `rootWin` because you can't skip rows or columns
- (g) [1 pt] You can create a frame inside another frame.
- A. **True**
  - B. False
- (h) [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. (4 points)

For each of the following multiple choice questions, indicate the single most correct answer by circling it!

- (a) [1 pt] Indicate the string below that will be matched by the regular expression: `\d{3}(\d{3})*[a-w]*`
- A. 123xyz
  - B. **223abc**

- C. 1236abc  
 D. 1237857abcde  
 E. 998 753 423ghij
- (b) [1 pt] Which of the following regex expressions will match entirely:  
 P. Sherman, 42 Wallabee Way, Sydney
- A. `\D+\d{2}?`  
 B. `\D+\d{2}?\D`  
 C. `[A-Z]+\S\s(\d{2}\s)?`  
 D. `(?:(?:[A-Za-z]+\S\s)|\d{2})+`
- (c) [1 pt] Which of the following regexes does **NOT** match the following string:  
 2012-12-01
- A. `2012-12-01`    B. `\d{4}-\d{2}-\d{2}`    C. `(?:\d{2,4}-?){3}`    D. `[0-2]{3,10}`  
 E. ....?
- (d) [1 pt] Which of the following input matches this regular expression (the last set of brackets includes a space after the period):  
`123\d{3}[A-Z][a-z]+ [A-Z][a-z\ . ]*`
- A. 1233 Main St. Avenue  
 B. 12398 techwood Drive N.  
 C. **123 N Bobby Dodd**  
 D. 123 SW Atlantic Drive

3. (9 points)

The Heat table:

| Number | Player   | Position | FGpct | PPG  | FTpct |
|--------|----------|----------|-------|------|-------|
| 6      | James    | SF       | 0.49  | 25.9 | 0.78  |
| 3      | Wade     | SG       | 0.4   | 15.9 | 0.75  |
| 1      | Bosh     | C        | 0.46  | 15.9 | 0.75  |
| 34     | Allen    | SG       | 0.43  | 10.2 | 0.87  |
| 15     | Chalmers | PG       | 0.42  | 9.4  | 0.75  |
| 11     | Anderson | PF       | 0.81  | 6.4  | 0.73  |
| 30     | Cole     | PG       | 0.48  | 6.1  | 0.74  |
| 40     | Haslem   | PF       | 0.59  | 5.1  | 0.57  |
| 31     | Battier  | SF       | 0.29  | 4.7  | 0.82  |

Using the database table **Heat** listed above, solve the following problems.

- (a) [3 pts] Write an SQL statement that will return each Player (name), their Number, and their FGpct, sorted by FGpct first, then number (in the case of a tie), with the smallest numbers first.

**Solution:**

```
SELECT Player, Number, FGpct FROM Heat ORDER BY FGpct, Number
+1 for SELECT Player, Number, FGpct
+1 for FROM Heat
+1 for ORDER BY FGpct, Number
```

- (b) [3 pts] Write an SQL statement that will return the number and name of all players with a position name that starts with an "S" (e.g. SG or SF in this example), sorted by player number with the largest number first.

**Solution:**

```
SELECT Number,Player FROM Heat WHERE Position LIKE 'S%' ORDER BY Number DESC
+1 for SELECT Number,Player FROM Heat
+1 for WHERE Position LIKE 'S%'
+1 for ORDER BY NUMBER DESC
```

- (c) [3 pts] Write an SQL statement that will change every player's FGpct and FTpct values to 0.00 if their PPG (Points Per Game) is less than 10:

**Solution:**

```
UPDATE Heat SET FGpct=0.00,FTpct=0.00 WHERE PPG < 10
+1 for UPDATE Heat
+1 for SET FGpct=0.00,FTpct=0.00
+1 for WHERE PPG < 10
```

## 4. (16 points)

You are tasked with designing a simple GUI that allows a customer to order a car. Cars come in either blue or red colors. Your GUI must have the following elements:

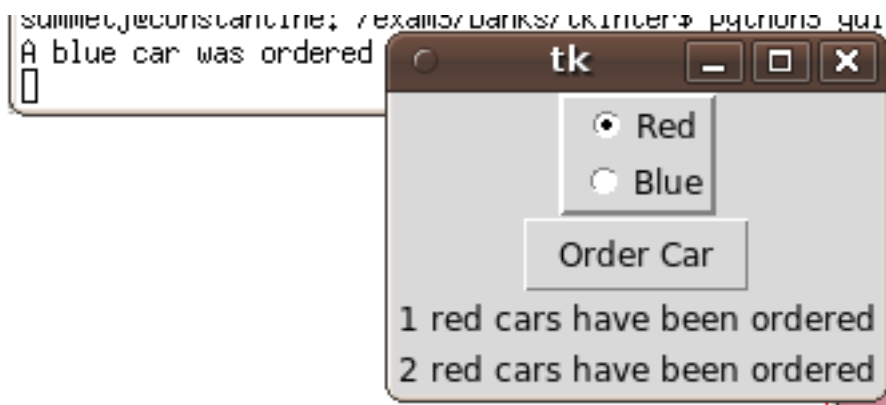
A “raised” frame with a border that is 2 pixels wide that contains two radio buttons. The two radio buttons have “Red” and “Blue” as their text. Neither should be selected by default. An “Order Car” button at the bottom of the window (NOT in the frame).

When the “Order Car” button is pressed, check to see which radio button is selected. If no radio button is selected, pop up a warning dialog telling the user that they need to select a color (and take no other action). If the Blue radio button is selected, PRINT (to the shell) “A blue car was ordered”. If the Red radio button was selected, add a label to the bottom of the GUI that says “X red cars have been ordered” (Where X is replaced with the actual number of times the user has pressed the “Order Car” button with the red radio button selected).

Here is a picture of the GUI before the button is pressed:



Here is a picture of the GUI (and shell) after the “Order Car” button is pressed 3 times (1 time with the “blue” radio button selected, 2 times with the “red” radio button selected).



Write python/tkinter code that would produce the GUI on the **next page**.

Write your python/tkinter code for the Car Ordering GUI here:

**Solution:**

```
from tkinter import *
from tkinter import messagebox

class CarGUI:
 def __init__(self, aWin):
 self.win = aWin
 self.numRed= 1
 f = Frame(aWin, relief='raised', border=2)
 f.pack()

 self.iv = IntVar()
 self.iv.set(0)

 Radiobutton(f, text="Red", value=1, variable=self.iv).pack()
 Radiobutton(f, text="Blue", value=2, variable=self.iv).pack()

 Button(aWin, text="Order Car", command=self.clicked).pack()

 def clicked(self):
 if self.iv.get() == 0:
 messagebox.showwarning("No Color!", "Please select a color!")
 return

 if self.iv.get() == 2:
 print("A blue car was ordered")
 elif self.iv.get() == 1:
 textStr = "{} red cars have been ordered".format(self.numRed)
 Label(self.win, text=textStr).pack()
 self.numRed= self.numRed + 1

win = Tk()
app = CarGUI(win)
win.mainloop()
```

**Grading:**



```

+1 point for importing tkinter correctly
+3 points for creating the frame, with border width = 2, and raised relief.
+2 points for creating the radio button with "Red" text inside the frame.
+2 points for creating the radio button with "Blue" text inside the frame.
+1 point for creating the "Order Car" button in the window, but NOT in the frame.

+2 points for popping up a "no color" warning if neither RB is selected.
+1 point for printing "A blue car was ordered" if the blue RB is selected.
+2 point for adding a label (to the bottom of the GUI) if the red RB is selected.
+1 point for putting a number into it every time.
+1 point for adding one to the number each time.

Misc minuses:
-1 point for not running a mainloop to create the gui.
-1 point for adding a widget to a non-existent container (non defined variable).

```

## 5. (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