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 6 questions on 12 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. Vocabulary	6	
2. SQL Syntax	3	
3. Baseball	13	
4. Regular Expressions	12	
5. countSmiles	9	
6. GUI coding	16	
Total:	59	

1. *(6 points)*
For each of the following vocabulary terms, write a concise 1-2 sentence definition. Be brief, and to the point.

(a) [3 pts] object

Solution: A compound data type that is often used to model a thing or concept in the real world. It bundles together the data and the operations that are relevant for that kind of data. Instance and object are used interchangeably. (An object whose type is of some class.)

Looking for: Compound Data Type, Includes data & operations/functions/methods.

(b) [3 pts] class

Solution: A user-defined compound type. A class can also be thought of as a template for the objects that are instances of it.

Looking for: Compound Data Type , can specify functions and group data, used to instantiate objects.

2. *(3 points)*
What is wrong with the syntax or logic of this SQL statement? Assume that "mydatabase" is a valid database name, and that "myRow" is a valid field (column) name. The SQL statement is trying to delete all records where the myRow entry has "prefix" at the beginning.

```
DELETE FROM mydatabase where myrow = prefix%
```

Circle **all** correct answers below:

A. There should be an asterisk after DELETE
B. "mydatabase" should be a column name instead
C. "mydatabase" should be a table name instead
D. "where" must be capitalized

E. The R in "myrow" must be capitalized
F. There should be quotes around 'prefix%'
G. The equals sign should be a LIKE

Solution: Correct answers: C F G
Grading: +1 point for each correct circle. -1 point for each incorrect circle. May not go below 0 points for the problem.

3. *(13 points)*

A table has been created for you with the following command:
CREATE TABLE baseball (Name TEXT NOT NULL, Number INTEGER, Hits INTEGER, Errors INTEGER)

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

The baseball table:

Name	Number	Hits	Errors
Caleb	11	5	2
Kibby	7	10	6
Moorissa	10	6	4

(a) [4 pts] Write an SQL statement that would add Daniel Palka into the table. His number is 29, he has 3000 hits, and -3 errors.

Solution: INSERT INTO baseball (Name, Number, Hits, Errors) VALUES ("Daniel Palka", 29, 3000, -3)

Grading:
+1 for INSERT INTO baseball
+1 for column/field names (in any order) (could also omit!)
+1 for VALUES
+1 for data in same order as column names (or in same order as table!)!

(b) [3 pts] Write an SQL statement that would change Caleb's hits to 7.

Solution: UPDATE baseball SET Hits=7 WHERE Name = 'Caleb'

Grading:
+1 for UPDATE baseball
+1 for SET Hits=7
+1 for WHERE Name = 'Caleb' (or Name LIKE 'Caleb')

(c) [3 pts] Write an SQL statement that returns the name and number of all players with more than 30 hits.

Solution: `SELECT Name,Number FROM baseball WHERE Hits > 30`

Grading:
+1 for SELECT Name,Number
+1 for FROM baseball
`+1 for WHERE Hits > 30 (or >= 31)`

(d) [3 pts] Write an SQL statement that returns the average number of hits for all players. Name the returned column "AVGHITS".

Solution: SELECT AVG(Hits) AS AVGHITS FROM baseball
Note: could also do SUM(Hits)/ COUNT(Hits)
Grading:
+1 for SELECT AVG(Hits)
+1 for AS AVGHITS
+1 for FROM baseball

4. *(12 points)*

For each regular expression given below, determine which of the lines that follow it are completely matched by the regex pattern. (That is, there are no characters in the string that will not be matched by the given pattern...) Circle **all** lines that are fully matched.

(a) [2 pts] Pattern: `$-?\d*.\d+`

$-.1301
$1
$-42+9001
$2316
$3.14

Solution: No patterns are matched. A $ in a regex has special meaning. If you want to match an actual $ you must use a backslash to escape it! Grading: +2 for 100% correct. Minus 1 for each extra or missing circle, minimum score of zero on this part.

(b) [2 pts] Pattern: `\d{3,5}\.\d?`

342.34
342.3
342q
342.
34212.4

Solution: 342.3
342.
34212.4
Grading:
+2 for 100% correct. Minus 1 for each extra or missing circle, minimum score of zero on this part.

(c) [2 pts] Pattern: `[A-Z]\S+\s[A-Za-z]*`

Bob Smith
bob smith
Jay W Summet
Keith O'hara
Julie Q
P D
Pq D
C3P0 z

Solution: Bob Smith
Julie Q
Pq D
C3P0 z

Grading: +2 for 100% correct. Minus 1 for each extra or missing circle, minimum score of zero on this part.

(d) [6 pts] Write a regular expression that will match a string formatted as: "NUM Some Street Name City, ST ZIPCO". The city name can contain any letters but will NOT have multiple words. It will always be followed by a comma and a space before the state abbriviation. The state abbriviation is exactly two upper case letters, followed by a space, followed by exactly 5 digits. Add 2 capturing groups to your RegEx that will capture (return) the street number and zip code ONLY! Example data:
555 8th Street Northwest Atlanta, GA 30332
50 Main Street Duluth, GA 30096
9 Gadsen-Heck Drive Duluth, GA 30096
333489 Georgia Tech Avenue Atlanta, GA 30332

Solution: One possible solution:
`(\d+)\s[^,]*,\s [A-Z]{2}\s([0-9]{5})`
Grading:
+1 for matching the digits of the street number
+1 for a street number capturing group.
+2 for correctly absorbing everything between the street number and zip code. Half credit if only minor change needed to make it right (e.g. only handling 3 words maximum street names). (Note that street names can have numbers, dashes, and spaces in them!)
+1 for matching exactly 5 digits in the zipcode.
+1 for a zip code capturing group.

5. *(9 points)*

You are to write a function named countSmiles 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 containing the number of times a smiley occurs. A Smiley is the two character combination of a colon and close parenthesis :).

Solution:

```
import urllib.request
from re import findall

def countSmiles(website):
    response = urllib.request.urlopen(website)
    html = str(response.read())
    data = findall(":\)", html)
    return len(data)
// Or, if they didn't use regular expressions, they could do this:
   counter = 0
   index = html.find(":)")
   while index != -1:
      index = html.find(":)",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 smiles (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

6. *(16 points)*

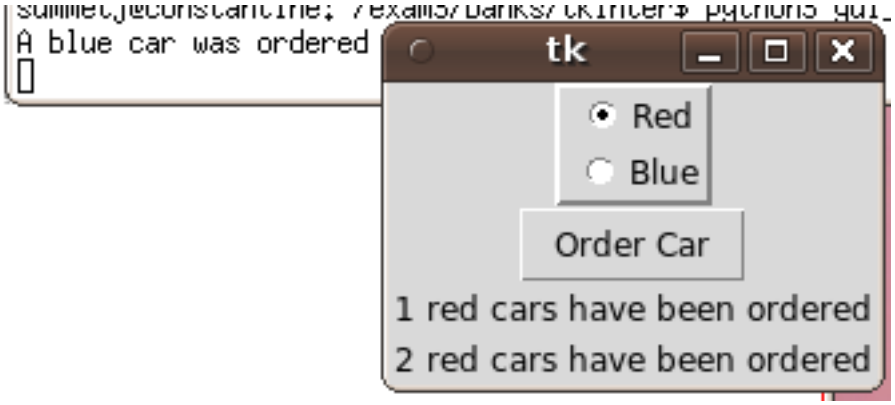
You are tasked with designing a simple GUI that allows a customer to order a car. Cars come in either blue or red colors. Your GUI must have the following elements:

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-existant container (non defined variable).
```

This page intentionally left blank. You may use it for scratch paper. If you place an answer on this page, box it, indicate which problem it is for by number, and BE SURE TO WRITE "Answer on last page" at the problem location!