

CS 1803 Spring 2011 Exam 3 KEY

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: _____

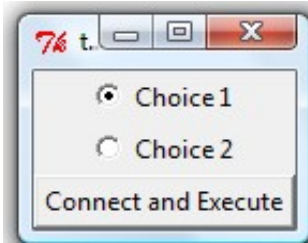
Question	Possible Points	Earned Points	Graded By
GUI	22		
SQL	24		
XML	20		
Multiple Choice	14		
Total Points	80		

GUI (22 Points)

Draw the GUI generated when the following code is run (be sure to specify if any radio buttons are selected) and **write the code** outlined by the comments in the update method. **Don't forget to draw the GUI!**

```
from tkinter import *
import pymysql
class testGUI():
    def __init__(self, master):
        self.root = master
        self.frame = Frame(self.root)
        self.frame.pack()
        self.radios()
        self.button = Button(self.root, text = "Connect and Execute", command = self.update)
        self.button.pack()
    def radios(self):
        self.choice = IntVar()
        r1=Radiobutton(self.frame, text="Choice 1", variable = self.choice, value = 0)
        r1.grid(row=0,column=0)
        r2 =Radiobutton(self.frame, text = "Choice 2", variable = self.choice, value = 1)
        r2.grid(row=1,column=0)
    def entries(self):
        self.entryStr = StringVar()
        entryBox = Entry(self.frame, textvariable=self.entryStr)
        entryBox.grid(row=2,column=0)
    def update(self):
        query1 = "SELECT SUM(demand) FROM demand WHERE name = 'Apple'"
        query2 = "SELECT AVG(demand) FROM demand WHERE name = 'Apple'"
        #write the code to connect to a database named cs1803_ie, on academic-mysql.cc.gatech.edu,
        # with user cs1803_ie and password "mypass"
        db = pymysql.connect(host = "academic-                                +3 connect correctly
            mysql.cc.gatech.edu",user="cs1803_ie",passwd="q8tRegu5", db="cs1803_ie")
        #write the code to execute query1 if the selected radio button is choice1
        #or query2 if the selected radio button is choice2
        #then print the result of the query to the shell using the print function.
        cursor = db.cursor()                                           +3 make the cursor
        if self.choice.get() == 0:                                     +2 get the radio button value
            cursor.execute(query1)                                     +3 execute the correct query
        elif self.choice.get() == 1:
            cursor.execute(query2)
        for item in cursor:                                           +3 print only the desired data
            print(item[0])
        cursor.close()                                               +3 close the cursor
                                                                    +5 GUI drawn correctly

root = Tk()
test = testGUI(root)
root.mainloop()
```



SQL (24 Points)

The student Table:

```
CREATE TABLE student
(id INTEGER PRIMARY KEY AUTO_INCREMENT UNIQUE NOT NULL,
name TEXT NOT NULL,
major TEXT,
gpa DECIMAL,
hours INT(10))
```

id	name	major	gpa	hours
1	"Jane Doe"	"ISyE"	3.25	15
2	"John Doe"	"CS"	2.97	18

All SQL statements you write must provide correct output and be syntactically correct.

Grading in general: -1 wrong conditional, -1 syntax error, -2 forgetting condition

Question 1: Write an SQL statement that will select the name and GPA of all students where the students' major is "CS".

```
SELECT name, gpa FROM student WHERE major = "CS"
```

Question 2: Return everything about a student who has 'Doe' in the name and who is a CS major.

```
SELECT * FROM student WHERE name LIKE "%Doe%" AND major = "CS"
-2 for forgetting %, -2 for using = instead of LIKE
```

Question 3: Add a new student with your name, major, a GPA of 4.00 and 17 hours; do not specify an ID.

```
INSERT INTO student (name, major, gpa, hours) VALUES ("Will Barr", "CS", 4.00, 17)
-2 for specifying the ID.
```

Question 4: Remove any student whose gpa is below 2.00 or whose hours are below 12.

```
DELETE FROM student WHERE gpa < 2.00 OR hours < 12.
```

Question 5: Update all ISyE majors who have more than 15 hours to have a GPA of 4.00.

```
UPDATE student SET gpa = 4.00 WHERE major = "ISyE" AND hours > 15
```

Question 6: Return the number of rows that have a major of "ISyE" or "CS"

```
SELECT COUNT(*) FROM student WHERE major = "ISyE" OR major = "CS"
```

Code Writing: XML (20 Points)

Given the xml file below, write a function that takes a single parameter named calculateCost. The parameter will be a file name to open. Your calculateCost function should calculate and return the total cost of purchasing all of the items listed in the file (this value will be the sum of the terms (#in stock * price per item)). Your code must work with a file of any length, in the general format of the fruits.xml example below.

While parsing the XML file, use a print() function to print the name of each product to the shell to assist in debugging.

Your final returned answer should be a float that is the cost in dollars For example, in the below file the "apple" contribution to the overall cost is \$60 (\$3 for each of the 20 apples), and the total sum of all products is $20 * 3 + 11 * 5.99 + 3 * 18.95 + 50 * 1.50 = 257.74$

fruits.xml:

```
<products>
  <product name="apple">
    <inStock>20</inStock>
    <price type="dollars">3.00</price>
  </product>
  <product name="pineapple">
    <inStock>11</inStock>
    <price type="dollars">5.99</price>
  </product>
  <product name="watermelon">
    <inStock>3</inStock>
    <price type="dollars">18.95</price>
  </product>
  <product name="grape">
    <inStock>50</inStock>
    <price type="dollars">1.50</price>
  </product>
</products>
```

```
import xml.etree.ElementTree as etree          +1 - import statement
def calculateCost(filename):                  +2 - method header
    tree = etree.parse(filename)              +2 - create the tree
    root = tree.getroot()                    +2 - get the root of the tree
    money = 0
    for product in root:                     +3 - loop/parse the entire file
        print("Product is: ",product.attrib["name"]) +2 - print the name of the product
        stock = float(product[0].text)        +2 - retrieve stock and price
        price = float(product[1].text)        +1 - cast stock and price as floats
        money = money + stock*price           +2 - calculate the contributing cost of each element
                                                +2 - calculate the total cost
    return money                              +1 - return statement
```

Multiple Choice (14 Points)

The following questions in this section may contain multiple correct answers. For those which do contain multiple correct answers, select all correct answers. *The letter for all selected answers must be written in the answer blanks provided in alphabetical order!* 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. We are nice, so you cannot earn below zero points on any one question.

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

1. AD 5. BCD

2. C 6. BD

3. BC 7. B

4. AB 8. D

- Which of the following statements is/are TRUE?
 - Every SELECT statement must have a FROM clause.
 - Every SELECT statement must have a WHERE clause.
 - You can only have 1 SELECT statement per query.
 - SELECT * returns data for all of the columns in the specified table
- Which of the following statements is/are TRUE?
 - A class can only be extended by 1 class (each parent-class can only have one subclass)
 - A subclass is created by doing a deep copy of the parent class.
 - A subclass can have methods with the same name as a method in the parent-class
 - A subclass only inherits from the immediate parent-class
- Which of the following is/are valid SQL functions that work on the class MySQL server?
 - AVERAGE()
 - STDDEV_SAMP()
 - MIN()
 - MODE()

Use the following code for questions 4 and 5:

```
class Student:
    def __init__(self, name, age, major = 'UEC'):
        self.name = name
        self.age = age
        self.major = major
        school = 'Georgia Tech'
```

- Which of the following create(s) an instance of the Student class?
 - s1 = Student('Joy', 20)
 - s2 = Student('Bob', 21, 'ISYE')
 - s3 = Student()

5. Which of the following statements is/are TRUE given the code above?
 - a) school can be accessed directly in other methods within the Student class
 - b) self.name can be accessed directly in other methods within the Student class
 - c) school is a local variable
 - d) self.name is an instance variable

6. When creating a GUI which of the following steps are MANDATORY?
 - a) build the GUI in a class
 - b) import the tkinter module
 - c) make a Frame widget to put the other widgets in
 - d) create an instance of the Tk class to be the root window

7. What are the 5 steps, in order, to read data from a website?
 - a) open the web-page, read the web-page, convert the data to a string, parse the string to retrieve data, close the web-page
 - b) import urllib.request, open the web-page, read the web-page, convert the data to a string, parse the string to retrieve data
 - c) import urllib.request, open the web-page, parse the html to retrieve the data, convert the desired data to a string, print the data
 - d) import urllib.request, read the web-page, convert the data to a string, parse the string to retrieve data, close the web page

8. Given a valid database db, what is the purpose of calling db.commit()?
 - a) disconnect from the database
 - b) to be able to run queries on the database
 - c) close the database
 - d) save any changes that have been made to the database