CS 1803 Practice Exam 3

Approximate point values before each question. This practice exam is slightly longer than your actual exam. Note that any material on the previous exams are also fair exam questions for your actual exam 3. Take this exam on a piece of paper, then check your answers using your computer.

Use the following Table PEOPLE to answer the next 4 questions

id	name	address	school	email
1	Jay Summet	102 Summet Drive	Georgia Tech	summetj@gatech.edu
2	Dr. Who	555 Science Street	Harvard	timetravel@harvard.edu
3	George Burdell	598 Legend Lane	Georgia Tech	georgepbudell@gatech.edu
4	John Doe	111 Generic Circle	UGA	random@uga.edu
5	John Doe	321 Blastoff Boulavard	Georgia Tech	whoisthis@gatech.edu

(2) Write an SQL Query that prints the name and email of every person whose last name is "Doe".

select name,email from people where name like '%doe'

(2) Write an SQL Query that changes the address of any record with the name "Jay Summet" to "6464 Atlantic Drive".

update people set address="6464 Atlantic Drive" where name="Jay Summet"

(2) Write an SQL Query that prints out the every name in the database alphabetically (starting with a,b,c,...). Do not print out multiple copies of the same name.

select distinct name from people order by name

(2) Write an SQL Query that deletes every entry in the database that does not have a school name of "Georgia Tech"

delete from people where school not like "Georgia Tech"

Use the following Table SALES to answer the next 3 questions

id	product	cost	sale price	date	items sold
1	Sham Wow	10	19.99	11-15-2010	11
2	Snuggie	13	24.5	11-15-2010	30
3	Swifter Duster	4	11.1	11-16-2010	9000
4	Snuggie	13	24.5	11-16-2010	22
5	Snuggie	13	24.5	11-17-2010	1
6	Sham Wow	10	19.99	11-17-2010	18
7	Cami Lace	8	16.75	11-17-2010	65

(3) Write an SQL Query that prints out the total items sold in the database for each day.

SELECT SUM(items sold) FROM sales GROUP BY date

(3) Write an SQL Query that prints out the profit (the differenece between the sale price and cost of the product) and product of each product without repeating any product more than once.

SELECT product, (sale price-cost) FROM sales GROUP BY (sale price-cost)

(4) Write an SQL Query prints out the total amount of money earned each day(the profit for every item sold that day) in descending order.

SELECT date, SUM((sale price - cost) *items sold) FROM sales GROUP BY date ORDER BY SUM((sale price-cost) *items sold) DESC

(5) Write a function named convertMoney that will convert a string version of any currency(i.e. dollars, euros, yen, etc). This function will take in a single string parameter and return a float version of the currency. For instance if convertData("$100.55") was called, the return value would be 100.55 as a float. You may assume that the value passed in is in that currency's correct format.

```
def convertMoney(strCurrency):

    Money = ''

    for letter in strCurrency:
        if letter in "0123456789.":
            Money = Money + letter


    return float(Money)
```

(10) Write a function named downloadHTML that will go to a webpage, read the html, and write that html out to a file. The function will take in two string parameters, the web address to the website and the name of the file that the user wants to create. Note: if you use any modules remember to import them! You may assume the web address will be a valid URL.

```
import urllib.request

def downloadHTML(address, fileOut):

    response = urllib.request.urlopen(address)
    html = response.read()
    text = str(html)

    file = open(fileOut, 'w')
    file.write(text)
    file.close()
```

Use the following code to answer the next 2 questions.

```
class Pokemon:

    def __init__(self, theType = 'normal', health = 10):
        self.type = theType
        self.health = health

    def struggle(self):
        print(" has used struggle!")

    def tackle(self):
        print("The pokemon has used tackle")


class Charmander(Pokemon):

    language = "char char charmander!"

    def __init__(self):
        super().__init__('water', 10)
        self.type = 'fire'

    def speak(self):
        print(self.language)

    def struggle(self):
        print("Charmander", end = '')
        super().struggle()


class Charmeleon(Charmander):

    health = 50

    def __init__(self):
        super().__init__()
        self.language = "charmeleon!"


pokemon1 = Charmander()
pokemon2 = Charmeleon()
```

(15) What is printed to the screen after the calls below?

pokemon1.speak()
pokemon1.struggle()

pokemon2.speak()

pokemon2.struggle()
pokemon2.tackle()
print(pokemon2.health)
print(pokemon2.type)

```
char char charmander!
Charmander has used struggle!
charmeleon!
Charmander has used struggle!
The pokemon has used tackle
10
fire
```

(10) Write a child class that inherents from the class Charmeleon that changes its language to "Charizard!", updates its health to 100, and updates the speak method print out the message twice.

```
class Charizard(Charmeleon):

    def __init__(self):
        super().__init__()
        self.language = "Charizard!"
        self.health = 100

    def speak(self):
        super().speak()
        super().speak()
```

(25) You have been hired by GT housing to write a program that will assign students to

rooms based on alphabetical order by their first name. Your job should you choose to accept it is to write a function named orderResidents which takes in a list students names in form ["First Name Last Name", ...] and returns a dictionary consisting of the the string "Room Number" as the key and the students assigned to that room as the value (e.g. print (residentHall["Room 1"]) should display the names of the residents in that room). Each room can hold up to 4 students and there are 50 rooms. If there are not enough students left to fill a room, insert the remaining students in the room. If there are no students assigned to the room, a value of None should be associated with that room. HINT: store the residents to be inserted into the correct room in a compund data type.

```
def orderResidents(array):

    rooms = {}
    residents = []
    array.sort()

    for room in range(50):
        rooms["Room " + str(room)] = None

    count = 0
    for num in range(0, len(array), 4):
        count += 1
        try:
            residents.append(array[num])
            residents.append(array[num + 1])
            residents.append(array[num + 2])
            residents.append(array[num + 3])
        except:
            pass
        rooms["Room " + str(count)] = residents
        residents = []

    return rooms
```

(35) Your employer has asked you to write a generalized program to download a table from his company's database and to write out the data from the table into a csv file. He then tells you because the database contains sevel thousand entries, that he only wants the top half of the table to be written out to keep the csv file manageable. After accepting this task, you have decided to split this major project into 4 seperate pieces so that a later employee may edit your code easily. Part 1 consists of writing a function to connect to a database, part 2 consists of writing a function that gets all the data from the table and converts it to a list, part 3 consists of writing a function to count the number of entries in the database, and part 4 consists of writing a function to take the data and count number found with the previous two functions and write out the first half of the table into a csv file. You may assume valid parameters are passed into these functions. HINT: Use the pymysql module to connect to a database through python.

PART 1 (5)
Write a function named connectToDatabase that takes in a username and password as parameters and returns the connected database object. (use "academic-mysql.cc.gatech.edu" as the host name and "cs1803_ie" as the database name).

```
#Part 1
def connectMySQL(username, password):

    db = pymysql.connect( host = "academic-mysql.cc.gatech.edu",
                          passwd = password, user = username, db='cs1803_ie')

    return(db)
```

PART 2 (10)
Write a function named downloadTable that takes in a table name and database as parameters and returns a list that includes all of the data from that specific table of the database. This should be done using SQL.

```
#Part 2
def downloadTable(table, db):

    cursor = db.cursor()
    cursor.execute("SELECT * FROM %s" % (table) )
    data = []
    for stuff in cursor:
        data.append(stuff)

    cursor.close()
    db.commit()

    return data
```

PART 3 (10)

Write a function named countEntries that takes in a table name and database object as a parameter and returns the number of rows (entries) of the database as an integer. This should be done using SQL.

```
#Part 3
def countEntries(table, db):
    cursor = db.cursor()
    cursor.execute("SELECT COUNT(*) FROM %s" % (table) )

    for item in cursor:
        count = item[0]

    cursor.close()
    db.commit()

    return count
```

PART 4

Write a function named writeCSV that takes in a list of data from the database, an integer that corresponds to the number of entries in the database, and a filename for the csv file. This function should write out the data from the table and write out this data in csv format. Note: a header row is not required and having a comma after the end element of each row is also fine. Remember to close your file!

```
#Part 4
def writeCSV(data, count, fileName):

    file = open(fileName, 'w')

    for index in range(count//2):
        for item in data[index]:
            file.write(str(item) + ",")

        file.write("\n")

    file.close()
```