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 will not be tolerated. You are to uphold the honor and integrity bestowed upon you by the Georgia Institute of Technology.

1. Keep your eyes on your own paper.
2. Do your best to prevent anyone else from seeing your work.
3. Do NOT communicate with anyone other than a proctor for ANY reason in ANY language in ANY manner.
4. Do NOT share ANYTHING during the exam. (This includes no sharing of pencils, paper, erasers).
5. Follow directions given by the proctor(s).
6. Stop all writing when told to stop. Failure to stop writing on this exam when told to do so is academic misconduct.
7. Do not use notes, books, calculators, etc during the 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.

Signature: ___________________________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>Possible Points</th>
<th>Points Earned</th>
<th>Graded By</th>
</tr>
</thead>
<tbody>
<tr>
<td>True False</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Queries</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connect</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Objects</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Scraper</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>78</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 1: Indicate True or False for each statement below 12 Points

1. A compound statement such as a class definition MUST have something in its body. TRUE

2. When writing to an XML file, all attributes must be integers. F- STRINGS

3. In HTML an end tag is indicated by a backslash, ex: </body>. TRUE

4. The following is a valid XML document:
   <foo></foo>
   <bar></bar> FALSE - It has two root elements

5. ADD, SELECT, and UPDATE are all valid SQL keywords. FALSE

6. To subclass an existing class, you must provide self and the superclass as parameters ex:
   class childClass( self, superClass): FALSE - self reference is not required.

7. All object methods must accept a parameter named self as their first parameter. FALSE - The first parameter is named self by convention, but does not need to be.

8. A function is a segment of code that performs a specific operation while a method is a function that is part of an object (and accepts an extra first parameter that references the specific instance). TRUE

Use the following code for questions 9-12

class Rondo:
    def __init__(self, assists):
        self.assists = assists
        self.points = 0
        self.vertical = 38

9. self.assists and self.points are instance variables. TRUE

10. Self.assists and self.points can be accessed from other methods. TRUE

11. The __init__() method is called each time an instance of the class Rondo is created. TRUE

12. The following code creates an object of type Rondo with 10 assists:
    Rondo(10) TRUE-But it does not save a reference to it.
Question 2: Short-Answer: SQL Query Writing 20 Points

Below is a table, testTable, which represents messages sent between students in a messaging application. The first row is the header which gives the name of each column; every row after that is a sample data row. The id, priority, and timestamp columns are INT columns. (id autoincrements) The username, recipient, and message columns are TEXT columns.

<table>
<thead>
<tr>
<th>id</th>
<th>username</th>
<th>recipient</th>
<th>message</th>
<th>priority</th>
<th>timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>william.barr</td>
<td>neaster3</td>
<td>Hey, check the website for the workshop stuff for ...</td>
<td>3</td>
<td>20101116</td>
</tr>
<tr>
<td>2</td>
<td>neaster3</td>
<td>william.barr</td>
<td>Ok; I'll see you on Monday.</td>
<td>3</td>
<td>20101116</td>
</tr>
<tr>
<td>3</td>
<td>egarcia3</td>
<td>1803TA</td>
<td>Hi everyone, remember we need people to proctor this...</td>
<td>2</td>
<td>20101119</td>
</tr>
<tr>
<td>4</td>
<td>spujari3</td>
<td>jstudent5</td>
<td>Ok, I've found the answer to the question you asked...</td>
<td>3</td>
<td>20101117</td>
</tr>
<tr>
<td>5</td>
<td>adamliem</td>
<td>jstudent5</td>
<td>Yes, you can see what we're going to be doing during...</td>
<td>3</td>
<td>20101115</td>
</tr>
<tr>
<td>6</td>
<td>summjet</td>
<td>1803TA</td>
<td>Next week during workshop, you should cover XML related...</td>
<td>1</td>
<td>20101119</td>
</tr>
</tbody>
</table>

In the following statements, write the SQL query which will perform the specified action. You should not write any Python code for the following questions.

*Assume that the table is restored to the original (above) after each query that would have actually changed the contents, so each question below should be treated independently.

1. Return the text of every message that has either been sent or received by william.barr

   SELECT message FROM testTable WHERE recipient="william.barr" OR username = "william.barr"
   +1 for SELECT message
   +1 for FROM testTable
   +1 for WHERE recipient="william.barr"
   +1 for OR username="william.barr"

2. Return the count of how many messages were sent on 20101116 with a priority of 3.

   SELECT COUNT(id) FROM testTable WHERE priority=3 AND timestamp = 20101116
   +1 for SELECT COUNT( <any field>)
   +1 for FROM testTable
   +1 for WHERE priority=3
   +1 for AND timestamp = 20101116

3. Return the timestamp and the average of all messages' priority on each day. Order the result by the timestamp in descending order.

   SELECT timestamp, AVG(priority) FROM testTable GROUP BY timestamp ORDER BY timestamp DESC
   +1 for SELECT timestamp, AVG(priority)
   +1 for GROUP BY timestamp
4. Add a row to the table where the username is william.barr, the recipient is 1803TA, there is no message (NULL, not an empty message), the priority is 5, and the timestamp is 20101120.

```sql
INSERT INTO testTable (username, recipient, priority, timestamp) VALUES
("william.barr", "1803TA", 5, 20101120)
```

5. Reset every row in the table that has a timestamp of 20101115 to have a priority of 1.

```sql
UPDATE testTable SET priority=1 WHERE timestamp = 20101115
```

**Code Writing- DB Connect**

1. Write a function named `connect` that takes in a username and password as parameters. Your function should connect to a MySQL database using the appropriate module. (Use "testDatabase.gatech.edu" as the host name and "Test3" as the database name) and then print a message telling the user that the connection is complete. Finally return the database connection object. (6 points)

```python
def connect(username, password):
    dat = pymysql.connect( host = "testDatabase.gatech.edu", passwd = password, user = username, db = "Test3")
    print("Connected to the database!")
    return dat
```

**points taken off for using "pass" as a variable because it is a keyword in python**
Code Writing- Objects  

Create a class named Student. Each Student will have a name and a major. The default major for each Student will be “UEC”. The \texttt{\_\_init\_} method has one parameter, name, which will be a string. This method should assign the value of the instance variable \texttt{name} to this input parameter.

```python
class Student():
    def \_\_init\_(self, name):
        self.name = name
        self.major = "UEC"
        # could do major = "UEC" in the main class body instead, although that would create a class variable...
```

Create a class, CsStudent, that extends Student. Each CsStudent should have “CS” as their major and a default grade of 0. The CsStudent’s name will be given as an input parameter when the class is instantiated. You \textbf{MUST} call the super class \texttt{\_\_init\_}() method to assign the name.

There will be two additional methods in the CsStudent class- takeTest and getGrade.

The takeTest method will have no parameters, and when called it will change the CsStudent’s grade to a random integer between 0 and 100 (inclusive).

The getGrade method will return the CsStudent’s grade.

```python
import random

class CsStudent(Student):
    def \_\_init\_(self, name):
        super().__init__(name)
        self.grade = 0
        self.major = "CS"

    def takeTest(self):
        randGrade = random.randint(0,100)
        self.grade = randGrade

    def getGrade(self):
        # correct header.
```
return self.grade + 1 for correct return.
Code Writing – Web Scraper

Write a function named readHTML, that accepts a URL as a string as it's only parameter. Your method should read web page and return the word that immediately follows the first `<h2>` tag as a string. You may assume that there is a space character after the word which indicates the end of the word.

Once your method is complete, write the line of code to call it using http://www.nytimes.com/ as the url and assign the return value to the variable myHTML. The module import line of code is given for you.

import urllib.request

def readHTML(url):
    response = urllib.request.urlopen(url)
    html = response.read()
    strHTML = str(html)
    count = 0
    startKeep = None
    result = ""
    for count in range(len(strHTML)):
        st = strHTML[count:count+4]
        ch = strHTML[count]
        if st == "<h2>":
            startKeep = count + 3
        for ch in strHTML[startKeep:]:
            #print("in the loop")
            if ch == " ":
                break
            else:
                result = result + ch
    return result

myHTML = readHTML("http://www.nytimes.com/")

Alternate method:
def readHTML(url):
    response = urllib.request.urlopen(url)
    html = response.read()
    htmstr = str(html)
startPos = htmlstr.find("<h2>") + 4  # First the first <h2> tag, move past it
endPos = htmlstr.find(" ", startPos)  # Finds the first space after the <h2> tag.
word = htmlstr[startPos:endPos]  # Copy just the word we want.
return(word)

myHTML = readHTML("www.nytimes.com")