

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 6 questions on 6 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	9	
2. Creating a Child	6	
3. Multiple Choice	4	
4. SQL Sales	24	
5. Ant XML	8	
6. Soccer XML	18	
Total:	69	

## 1. (9 points)

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

(a) [3 pts] flow of execution

(b) [3 pts] parameter

(c) [3 pts] element (in XML)

## 2. (6 points)

Given the code below, write a python code example showing two DIFFERENT ways to create and attach a subelement to the root element. Use tag names of 'child1' and 'child2' for the two elements. Set the TEXT of child1 to be "Kid 1" and add an attribute of kidNumber="2" for child2.

```
import xml.etree.ElementTree as etree
root = etree.Element("MyRoot")
```

## 3. (4 points)

For each of the following multiple choice questions, indicate the most correct answer! Indicate your selected answer by circling it.

- (a) [1 pt] Which of the following ways will correctly place attributes in an XML Element? You may assume `etree` is the correct import.
- A. `myDict = {"key1": "value1"}`  
`element = etree.Element("tag", myDict)`
  - B. `element = etree.Element("tag", key1="value1", key2="value2")`
  - C. `element = etree.Element("tag")`  
`element.get("key1", "value1")`
  - D. `element = etree.Element("tag")`  
`element.set(value1 = "key1")`
- (b) [1 pt] Which of the following statements is correct of proper XML?
- A. A particular XML tree may possess only one root.
  - B. An element may have only one child.
  - C. Every element must have an opening and closing tag.
  - D. An element may not have an attribute named 'text'.
- (c) [1 pt] Given the following line of XML, what is the correct term for "mystery1"?
- ```
<mystery1 mystery2="mystery3">mystery4</mystery1>
```
- A. Tag
  - B. Attribute
  - C. Text
  - D. Element
- (d) [1 pt] Every element in an XML tree has at least one parent.
- A. True
  - B. False

## 4. (24 points)

A table has been created for you with the following command:

```
CREATE TABLE Sales ( Id INTEGER NOT NULL AUTO_INCREMENT, Name TEXT,
Email TEXT, Demand INTEGER, Order INTEGER )
```

- (a) [4 pts] Write the SQL command that will return (only) the name of all people who have an email address ENDING in .edu:
  
- (b) [3 pts] Write the SQL command that will return the calculated ratio between the Demand and Order integers (the ratio is calculated as Demand / Order ), naming the returned column "Ratio".
  
- (c) [5 pts] Write the SQL command that will return the name of each person in the table, along with the average demand for that person. (Each person may have multiple records, each with a different demand number.) Sort your results by the average demand in ascending order.
  
- (d) [2 pts] Write the SQL command that delete all records from the table.
  
- (e) [10 pts] Write PYTHON CODE that connects to the "db.example.com" database host, with the username 'cs2316' and password 'SECRET', accessing the 'cs2316db' database. Then, execute the following SQL query: "SELECT \* FROM Sales". Take the results you receive back from the database server and PRINT them to the screen, but OMIT the first (Id) column from your printed output!

5. (8 points)

Here is a simple XML file:

```
<tick>
  <hillInfo>
    <x>90</x>
    <y>43</y>
    <ants>4</ants>
  </hillInfo>
  <antInfo>
    <ant x='91' y='44'>F W W W F FA F FH</ant>
    <ant x='90' y='45'>F FA F F W EA EA W</ant>
    <ant x='89' y='50'>W W EA EA EA F EA W</ant>
  </antInfo>
</tick>
```

Assume that the filename variable correctly indicates the above XML file. Write exactly what is printed when the code below is executed:

```
import xml.etree.ElementTree as etree
tree = etree.parse(filename)
root = tree.getroot()

for child in root:
    print(child.tag)

for child in root.find("antInfo"):
    print(child.attrib["x"])

for child in root.find("hillInfo"):
    print(child.text)

for child in root.findall("ant"):
    print(child.attrib["y"])
```

## 6. (18 points)

Examine the following python code. Write out the textual XML file (soccer.xml) that is written to disk after it is executed. You must indent children under their parents to show the containment relationship (in addition to having the start and stop tags in the correct places.

```
import xml.etree.ElementTree as TR

soccerplayer =( ( ("Frank Lampard", "11"), ("Didier Drogba", "5") ),
                ( ("Sergio Aguero", "21"), ("David Silva", "6") ) )

root = TR.Element("soccerteam")

t1=TR.SubElement(root,"team2",name="Manchester City",rank=str(2))
t2 = TR.Element("team1", name="Chelsea",rank=str(6))

count=0
for ss in soccerplayer[0]:
    print(ss)
    TR.SubElement(t1, "player", number=str(count), score=ss[1]).text=ss[0]
    count= count+1

for ss in range(len(soccerplayer[1])):
    nameplayer=soccerplayer[1][0]
    p2=TR.SubElement(t2,"player",number=str(ss))
    p2.text=nameplayer[ss]
    TR.SubElement(t2,"goals",score=soccerplayer[1][1][ss])

root.append(t2)
tree = TR.ElementTree(root)
tree.write("soccer.xml","UTF-8")
```