

CS 2316 Exam 3

Practice

ANSWER KEY

- Failure to properly fill in the information on this page will result in a deduction of up to 5 points from your exam score.
- Signing signifies you are aware of and in accordance with the **Academic Honor Code of Georgia Tech** and that you will not discuss this exam with other students.
- Calculators and cell phones are NOT allowed.
- Answers containing Python code must use valid Python code, including case-sensitivity, syntax, and API correctness.

Question	Points per Page	Points Lost	Points Earned	Graded By
Page 1	15	-	=	
Page 2	15	-	=	
Page 3	30	-	=	
Page 4	25	-	=	
Page 5	25	-	=	
TOTAL	110	-	=	

1. **Multiple Choice** Circle the letter of the best answer.

[3] (a) Given this definition:

```
d = {
  "people": {
    "person": [
      {
        "firstName": "Alan",
        "lastName": "Turing",
        "professions": {
          "profession": ["Computer Scientist", "Mathematician",
            "Computer Scientist", "Cryptographer"]
        }
      },
      {
        "firstName": "Stephen",
        "lastName": "Hawking",
        "professions": {
          "profession": ["Physicist", "Comedian"]
        }
      }
    ]
  }
}
```

[3] (b) Which of the following returns the second profession of Stephen Hawking (whose value would be 'Comedian')?

- A. `d['people']['person'][1]['professions']['profession'][1]`
- B. `d['people']['person'][1]['professions']['profession']`
- C. `d['people']['person'][1]['professions']['Comedian']`

[3] (c) What's the type of `d['people']['person'][1]['professions']['profession']`

- A. tuple
- B. dict
- C. list

[3] (d) What's the value of `d['people']['person'][0]['firstName']`?

- A. 'Hawking'
- B. 'Stephen'
- C. 'Turing'
- D. 'Alan'

[3] (e) Which of the following Python expressions opens a file for reading as text?

- A. `open "season"`
- B. `open("borders", 'wb')`
- C. `open("sesame", 'r')`
- D. All of the above

2. **Multiple Choice** Circle the letter of the best choice.

- [3] (a) The fundamental data abstraction in relational databases is the table.
A. True
B. False
- [3] (b) In order for a foreign key in one table to reference a primary key in another table, it must have the same name.
A. True
B. False
- [3] (c) An author can write many books and a book can have many authors. What kind of cardinality relationship exists between authors and books?
A. many to many
B. one to one
C. one to many
- [3] (d) The CSV data model can encode any data model that the XML data model can.
A. True
B. False
- [3] (e) Which of the following is **not** well-formed XML?
A. `<a> c `
B. `<a> <c> `
C. `<a> <c/> `
D. `<a> <c> d </c> `

3. Short Answer

- [5] (a) What command would you type in iPython to find your present working directory?

```
pwd or %pwd
```

- [5] (b) How would you find out what the `%prun` command does in iPython?

```
%prun?
```

- [5] (c) Write an expression that creates a NumPy array of 5 integers. Assume `import numpy as np` has been done.

Many possibilities

```
np.arange(5)
```

```
np.zeros(5, dtype=int) # or np.ones
```

```
np.array([0, 1, 2, 3, 4])
```

- [5] (d) Write an expression that creates a 3 x 3 NumPy array of integers. Assume `import numpy as np` has been done.

Many possibilities

```
np.arange(9).reshape((3,3))
```

```
np.zeros(9, dtype=int).reshape((3,3))
```

```
np.array([[1,2,3],[4,5,6],[7,8,9]])
```

- [5] (e) Given a dictionary `d` created by `d = dict(zip(['a', 'b', 'c', 'd'], range(4)))`, write a statement that creates a Pandas Series from `d` and assigns it to the variable `data`. Assume `import pandas as pd` has been done.

Many possibilities

```
data = pd.Series(d)
```

- [5] (f) After creating the series `data` above, what would `data['b']` return?

```
1
```

4. Short answer

Given:

```
salary = {"Data Scientist": 110000,  
          "DevOps Engineer": 110000,  
          "Data Engineer": 106000,  
          "Analytics Manager": 112000,  
          "Database Administrator": 93000,  
          "Software Architect": 125000,  
          "Software Engineer": 101000,  
          "Supply Chain Manager": 100000}  
openings = {"Data Scientist": 4184,  
            "DevOps Engineer": 2725,  
            "Data Engineer": 2599,  
            "Analytics Manager": 1958,  
            "Database Administrator": 2877,  
            "Software Architect": 2232,  
            "Software Engineer": 17085,  
            "Supply Chain Manager": 1270}
```

- [5] (a) Write a statement that assigns to `salary_data` a Panda series with the data from the `salary` dictionary.
`salary_data = pd.Series(salary)`
- [5] (b) After the assignment above, what is the value of `salary_data[Software Engineer]`
101000
- [5] (c) Write a statement that assigns to `jobs` a Panda DataFrame from the data in the `salary` and `openings` dictionaries with 'salary' as the heading for the salary column and 'openings' as the heading for the openings column.
1 pt 1 pt 1 pt 1 pt (1 pt remainder of syntax)
`jobs = pd.DataFrame({'salary': salary_data, 'openings': openings})`
- [5] (d) Write an expression that returns all the jobs in the `jobs` DataFrame with salary greater than 100000.
`jobs[jobs[salary] > 100000]`
- [5] (e) Write an assignment statement that adds a column to `jobs` called '6 figures' whose values are `True` for jobs with salaries greater than 100000 and `False` otherwise.
2 pts 3 pts
`jobs[6 Figures] = jobs[salary] > 100000`

5. Short Answer

Assuming a database with the following schema is stored in an SQLite3 database file named `dorms.db`,

```
create table dorm (
    dorm_id integer primary key autoincrement,
    name text,
    spaces integer
);
create table stud (
    stud_id integer primary key autoincrement,
    name text,
    gpa float,
    dorm_id integer references dorm(dorm_id)
);
```

- [15] (a) write a snippet of Python code that queries the database and stores in a variable named `dorm_assignments` a list whose elements are tuples, where each tuple contains a student name and the name of the dorm that student lives in, e.g., tuples like `('Cartman', 'Armstrong')`. Assume the `sqlite3` module is imported.

Solution:

```
conn = sqlite3.connect("dorms.db")           # 2 pts
curs = conn.cursor()                         # 1 pt
curs.execute("select stud.name, dorm.name " + # 2 pts execute(), 3 pts select header
              "from student join dorm using(dorm_id)") # 4 pts correct join
dorm_assignments = curs.fetchall()          # 3 pts (other methods also correct)
```

- [5] (b) Write a single Python expression that creates a tuple mapping student names to the names of the dorms they live in using the `dorm_assignments` list created above.

Solution:

```
dict(dorm_assignments)
or
2 pts      2 pts      1 pt
{stud:dorm for stud, dorm in dorm_assignments}
or
{t[0]:t[1] for t in dorm_assignments}
```

- [5] (c) Write a single Python expressions that creates a list of students in Armstrong using the `dorm_assignments` list created above.

Solution:

```
1 pt      1 pt      1 pt      2 pts
[stud for stud,dorm in dorm_assignments if dorm == 'Armstrong']
or
[t[0] for t in dorm_assignments if t[1] == 'Armstrong']
```