CSE 503 - Final Exam

12/12/22 @ 11:45AM, Alumni 97

Name: ______________________________ Person #: ________________________

UBIT: ______________________________ Seat #: ________________________

Academic Integrity

My signature on this cover sheet indicates that I agree to abide by the academic integrity policies of this course, the department, and university, and that this exam is my own work.

Signature: _____________________________________ Date: ___________________________

Instructions

1. This exam contains **17 total pages** (including this cover sheet). Be sure you have all the pages before you begin.
2. Clearly write your name, UBIT name, person number, and seat number above. **Additionally, write your UBIT name at the top of every page now.**
3. You have 3 hours to complete this exam. Show all work where appropriate, but keep your answers concise and to the point.
4. After completing the exam, sign the academic integrity statement above. Be prepared to present your UB card upon submission of the exam paper.
5. You must turn in all of your work. No part of this exam booklet may leave the classroom.

DO NOT WRITE BELOW

M1	M2	M3	M4	Total
81	81	24	14	200

Module 1 - Python [81 points]

Question 1 [15 points]

For this question, reference the code below for all answers. The code is syntactically correct.

```
def countSequence(seq):
  dict = {}
  for x in seq:
    dict[x] = dict.get(x, 0) + 1
  return dict
```

For each of the following, provide a specific example from the reference code, or write "no example" if there is no example in the reference code. Be sure to include exactly what is asked for, no more, no less. The first one has been filled in for you as an example.

0. variable name

   `dict`

1. argument list [3 points]

   `x, 0 or (x, 0)`

2. expression [3 points]

   `dict.get(x, 0) + 1 (or 1, dict.get(x, 0), {}, dict, etc)`

3. conditional statement [3 points]

   `no example`

4. keyword [3 points]

   `def, for, in, or return`

5. assignment statement [3 points]

   `dict = {}, or dict[x] = dict.get(x, 0) + 1`

Question 2 [12 points]

For this question reference the code below for all answers. The code is syntactically correct.

```
def ohno(s, flip):
  sum = 0
  for x in s:
    if flip == True:
      x = x * -1
    sum = sum + x
  if sum < 0:
    print("You've lost " + str(x * -1) + " apples!")
  elif sum > 0:
    print("You've gained " + str(x) + " apples!")
  return sum
```

For each of the following function calls, state what value is returned (if any), and state what output is displayed (if any).

Technically there was a typo in the above code. It was intended to print out the value of the sum variable, not of x. Both versions were accepted, congrats to those that caught my mistake :)

1. `ohno([1,2,3,4], False)` [3 points]

   Return Value: **`10`**

   Output: **`"You've gained 10 apples!"`**

2. `ohno([1,-10], False)` [3 points]

   Return Value: **`-9`**

   Output: **`"You've lost 9 apples!"`**

3. `ohno([4, -16, 2], True)` [3 points]

   Return Value: **`10`**

   Output: **`"You've gained 10 apples!"`**

4. `ohno([12, -6, -6], True)` [3 points]

   Return Value: **`0`**

   Output:

Question 3 [14 points]

You are working for an online shopping company. Every order made has a base cost (in dollars), and a weight (in pounds). Additionally, some customers have a membership which offers them certain perks. The total cost of an order is computed as follows:

- total cost = base cost + shipping cost
- Normally, shipping cost is $0.50 per pound, however:
  - if the base cost is > $100 then the customer gets free shipping
  - customers that have a membership always get free shipping

Consider a function named `totalCost`, that will be called with three arguments: the first two are integers corresponding to base cost and weight, the third is a boolean argument that determines whether or not the customer has a membership.

Describe, in general, what a test case is: [4 points]

A test case is a specific input to a particular piece of code, along with the expected result/output of executing the code with that input.

Below, write two distinct test cases for the `totalCost` function: [10 points]

```
totalCost(100, 100, False) should return 150
totalCost(100, 100, True) should return 100

(or any other examples that roughly follow the above format)
```

Question 4 [18 points]

Define the `totalCost` function from the previous question in Python below:

```
def totalCost(b, w, m):
  if b > 100 or m == True:
    return b
  else:
    return b + w * 0.5
```

Question 5 [10 points]

```
def takeMeTo(filename):
  with open(filename, "r", newline='') as f:
    reader = csv.reader(f)
    for line in reader:
      print("Go " + line[1] + " miles " + line[0] + " " + line[2])

takeMeTo("funkytown.csv")
```

Give possible contents for the file `funkytown.csv` which would result in the following output:

```
Go 12 miles east quickly
Go 9 miles south with style
Go 32 miles west casually
```

Write your answer below:

```
east, 12, quickly
south, 9, with style
west, 32, casually
```

Question 6 [12 points]

Define a function in Python named `totalMilesTo` that takes the filename of a csv file with the same format you've described above, and returns the total number of miles in the trip described by the `takeMeTo` function. For example, `totalMilesTo("funkytown.csv")` should return `53`.

Write your answer below:

```
def totalMilesTo(filename):
  total = 0
  with open(filename, "r", newline="") as f:
    reader = csv.reader(f)
    for line in reader:
      total = total + int(line[1])
  return total
```

Module 2 - JavaScript/HTML [81 points]

Question 1 [15 points]

For this question, reference the code below for all answers. The code is syntactically correct.

```
function doMouseThings(grid) {
  let eaten = 0;
  for (let i of grid) {
    for (let j of i) {
      if (j.contents == "CHEESE") {
        eaten = eaten + 1;
        console.log("Yum!");
      }
    }
  }
  return eaten;
}
```

For each of the following, provide a specific example from the reference code, or write "no example" if there is no example in the reference code. Be sure to include exactly what is asked for, no more, no less. The first one has been filled in for you as an example.

0. string literal

   `"CHEESE"`

1. parameter list [3 points]

   `grid or (grid)`
2. a statement that results in output [3 points]

   `console.log("Yum!");`
3. an expression that evaluates to a Boolean value [3 points]

   `j.contents == "CHEESE"`
4. variable declaration [3 points]

   `console.log("Yum!");`
5. looping statement (the whole statement) [3 points]

```
for (let j of i) {
  if (j.contents == "CHEESE") {
    eaten = eaten + 1;
    console.log("Yum!");
  }
}
```

Question 2 [12 points]

For this question reference the code below for all answers. The code is syntactically correct.

```
function spooky(x, y, z):
  let steps = 0;
  for (let i = 0; i < z; i = i + x) {
    if (i < y) {
      console.log("...");
    } else {
      console.log("BOO!");
    }
    steps = steps + 1;
  }
  return steps;
}
```

For each of the following function calls, state what value is returned (if any), and state the number of times that "BOO!" is printed to the console.

1. `spooky(1, 5, 10)` [3 points]

   Return Value: 10

   "BOO!" Count: 5

2. `spooky(1, 0, 6)` [3 points]

   Return Value: 6

   "BOO!" Count: 6

3. `spooky(7, 5, 10)` [3 points]

   Return Value: 2

   "BOO!" Count: 1

4. `spooky(7, 2, -4)` [3 points]

   Return Value: 0

   "BOO!" Count: 0

Question 3 [18 points]

Define a function in JavaScript named `positiveTotal` that takes a list of numbers and returns the sum of just the positive numbers in the list. For example, `positiveTotal([-1, 4, -7, 20])` should return `24`. If there are no positive numbers in the list, the function should return `0`.

```
function positiveTotal(a) {
  let total = 0;
  for (let x of a) {
    if (x > 0) {
      total = total + x;
    }
  }
  return total;
}
```

Question 4 [10 points]

For questions 4-6, reference the HTML below.

```
<head>
</head>
<body onload="greet();">
  <h1 id="header"></h1>
  <div id="schedule"></div>
  <input type="button" onclick="getSched();">Click Me!</input>
<body>
```

Based on the above HTML, define the JavaScript function that is called when you first visit the webpage so that it sets the content of the `<h1>` element to a greeting based on the time of day. If it is before noon, the greeting should say `"Good Morning!"`. If it is between noon and 6PM, it should say `"Good Afternoon!"`. Otherwise it should say `"Good Night!"`.

For this function you can assume you have previously defined a function named `getHourOfDay()` that returns the hour of the day from 0 to 23 (noon is 12, and 6PM is 18).

For reference, the function to get an HTML element by id is: `document.getElementById(...)` The object returned by this function has the key `innerHTML`, and the value associated with that key is the content of the HTML element.

Write your answer below:

```
function greet() {
  let h = document.getElementById("header");
  if (getHourOfDay() < 12) {
    h.innerHTML = "Good Morning!";
  } else if (getHourOfDay() < 18) {
    h.innerHTML = "Good Afternoon!";
  } else {
    h.innerHTML = "Good Night!";
  }
}
```

Question 5 [10 points]

We want to set up our client server relationship such that when the button is clicked, an AJAX request is sent to the server to get your schedule for the day, which will then be displayed in the `<div>` element on the page. Define the JavaScript function that is called when the button is clicked so that it makes an AJAX request to the `"/get_schedule"` path of your server.

You can assume you have already defined the `ajaxGetRequest(path, callback)` as we have in class. For the callback, you can assume you have already defined a function named `displaySchedule(...)`.

Write your answer below:

```
function getSched() {
  ajaxGetRequest("/get_schedule", displaySchedule);
}
```

Question 6 [16 points]

Finally, we must define the `displaySchedule` function. As a reminder, this function will receive a JSON string as a response from the server, and we can turn that JSON string into JavaScript data using the `JSON.parse(...)` function.

The data will be a list of strings, and the `displaySchedule` function should change the content of the `<div>` element such that it shows each string on its own line.

As a reminder the HTML element for a line break is `<br>`.

Write your answer below:

```
function displaySchedule(response) {
  let data = JSON.parse(response);
  let div = document.getElementById("schedule");
  let content = "";
  for (x of data) {
    content = content + x + "<br>";
  }
  div.innerHTML = content;
}
```

Module 3 - Programming Concepts [24 points]

Question 1 [10 points]

For each item labeled (a) through (e), match it with its ***best*** description from items (0) through (9). Not all items from the numbered list will be used.

(a) JSON
(b) return statement
(c) SQL
(d) bottle route
(e) key value pair

(0) a way to uniquely identify an HTML element
(1) a language for interacting with databases
(2) a way for a client to send data to a server
(3) a way to request a particular resource from a web server
(4) a way to specify which function handles an HTTP request
(5) a way to display data to the screen
(6) a way to encode data values as strings
(7) a way to get code from a web server
(8) a way to specify the value returned by a function
(9) an association of one piece of data with another

(a) best matches with __6__ [2 points]

(b) best matches with __8__ [2 points]

(c) best matches with __1__ [2 points]

(d) best matches with __4__ [2 points]

(e) best matches with __9__ [2 points]

Question 2 [8 points]

Name the 3 constructs (in either Python or Java) we discussed in class which allow us to store values in our programs.

variable, list, dictionary (or variable, array, object in JS)

Briefly describe a specific use case for each construct you listed above.

variable: any logical answer which has a need to store a single value
list: any logical answer which has a need to store multiple values in a particular order
dictionary: any logical answer which has a need to store multiple key-value pairs

Question 3 [6 points]

Name 2 ways to make your code easier to read / understand:

Any of the following:

- Comments
- Good/logical names for variables, functions, etc
- Tests/assertions
- Split code into multiple functions
- Organize code across multiple files
- Anything else I didn't think of that made sense

Module 4 - Misc Concepts [14 points]

Question 1 [6 points]

In class we briefly discussed algorithmic complexity. If given the choice of three algorithms that all solve the same problem, and run in O(n) steps, O(log n) steps, and O(n^2) steps respectively, which algorithm would you choose?

O(log n)

Briefly explain why your choice is better than the other two options:

An O(log n) algorithm takes fewer steps than an O(n) and O(n^2) algorithm as the input grows in size.

Question 2 [8 points]

Briefly describe what an injection attack is, and what precautions should be taken to prevent them:

An injection attack is when user input is allowed to include code which modifies how our programs behave in unintended ways. To prevent them, user input must be properly sanitized.

Explain the difference between the following two SQL statements:

```
1. cursor.execute("SELECT * FROM tableName WHERE field = ?", (variable,))
2. cursor.execute("SELECT * FROM tableName WHERE field = " + variable)
```

1. uses SQLite to populate the query, which ensures the variable will be treated only as data not code.
2. uses Python string concatenation to build a query string, which can allow for injection attacks if the variable is not properly sanitized.
(we should always opt for option 1)

Scrap Paper