EXAMINATION INSTRUCTIONS

This examination has 17 pages. Let the course staff know if you are missing a page. You have 3 hours to complete this examination. Use your time accordingly.

READ AND OBSERVE THE FOLLOWING RULES:

► Names are pre-printed on the exam booklets. Ensure that you have YOUR exam.
► Sign, using your usual signature, in the space provided on the back cover.
► All of your writing must be handed in. This booklet must not be torn or mutilated in any way and cannot leave the examination room.
► For your answers to earn credit, you must show all of your work in arriving at an answer. Partial credit will be earned as appropriate.
► Students can talk to proctors only in cases of supposed errors or question ambiguity. Proctors will not answer any other questions during this exam.
► Candidates guilty of any dishonest practices will be immediately dismissed from the examination and shall be liable for disciplinary action. Examples of dishonest practices include:
  ♦ Having books, papers, calculators, computers, cellphones, or other memory aid or electronic device out without explicit authorization by the proctors.
  ♦ Speaking or communicating with other students.
  ♦ Purposely exposing their work where other students could view it. Claims that the exposure was accidental or due to forgetfulness will not be accepted.

--------------------------- DO NOT WRITE BELOW THIS LINE! ---------------------------
Module 1 - Question 1 [20 points, 4 points each]
The code given below is syntactically correct.

Circle, and identify by number, EXACTLY 1 example of each of the following items in the code below. If the code below does not contain an example of an item, write "no example" next to that item in the list. To show you how I want the question answered, the first one is done for you.

0. numeric literal
1. keyword
2. variable
3. assignment statement
4. function argument
5. str literal no example!

```python
def overspent ( budget ) :  
    over = {}  
    months = list( budget.keys() )  
    months.pop(0)  
    for key in months : 
        data = budget[key]  
        diff = int(data[ 0 ]) - int(data[ 1 ])  
        if (diff < 0) :  
            over[key] = diff
    return over
```

Sample
Module 1 - Question 2 [20 points total, 10 each part]

[ ] 10 points: perfect
[ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Study the following code, then answer the question which follows.

```javascript
function module1question2(x) {
  result = ''
  if (x < 55) {
    return "Zoo";
  } else if (x < 70) {
    result = "Walk";
    x = x - 55;
  } else if (x < 85) {
    result = "Talk";
    x = x - 70;
  } else {
    result = "Hawk";
    x = x - 85;
  }
  if (x < 5) {
    result = result + "??";
  } else if (x < 10) {
    result = result + "!!";
  } else if (x < 15) {
    result = result + "**";
  }
  return result;
}
```

[10 points] What does the following statement print?
```javascript
console.log("Answer is " + module1question2(71));
```

Write your answer below:

Answer is: Talk??

[10 points] Give a value of x which `module1question2(x)` returns "Hawk**". Write your answer below:

95, 96, 97, 98, or 99

(Any one of these is a correct answer. We only listed all of them here to avoid any confusion about what the range of legal answers are)
Module 1 - Question 3 [20 points total]

Read this description of how to convert a temperature from Fahrenheit to Kelvin:

Fahrenheit and Kelvin are two common temperature scales. The Fahrenheit scale is used in the United States, while Kelvin is an absolute temperature scale, used worldwide for scientific calculations. While you might think this conversion wouldn’t occur much, it turns out there is a lot of scientific and engineering equipment that uses the Fahrenheit scale! Fortunately, it is easy to convert Fahrenheit to Kelvin.

Fahrenheit to Kelvin Method #1

1. Subtract 32 from the Fahrenheit temperature.
2. Multiply this number by 5.
3. Divide this number by 9.
4. Add 273.15 to this number.

The answer will be the temperature in Kelvin.

Define a Python function named \texttt{f2k} which converts a temperature expressed in Fahrenheit to the equivalent temperature expressed in Kelvin, using the above method. Your function should take a temperature in Fahrenheit as its only input and return the corresponding temperature in Kelvin.

Write your answer below:

\begin{verbatim}
(Many solutions are possible, this is just one example of code you could write. The important thing is whatever code you write, it correctly solves the problem.)

def f2k(temp) :
    retVal = temp - 32
    retVal = retVal * 5
    retVal = retVal / 9
    retVal = retVal + 273.15
    return retVal
\end{verbatim}
Module 1 - Question 4  [20 points total]

Consider a function named \texttt{postage} which will be called with one argument (you can assume that it will be called only with an integer value representing the weight of a package, in grams). This function will compute and return an integer value representing the postage needed to mail a letter based upon its weight, according to this table:

<table>
<thead>
<tr>
<th>Weight (in grams)</th>
<th>Postage (in cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to and including 20 grams</td>
<td>10</td>
</tr>
<tr>
<td>More than 20 grams, but less than or equal to 40 grams</td>
<td>$10 + 2$ cents per gram of weight over 20 grams</td>
</tr>
<tr>
<td>More than 40 grams</td>
<td>$50 + 3$ cents per gram of weight over 40 grams</td>
</tr>
</tbody>
</table>

Describe briefly what a test case is: [ 8 points total ]

[ ] 8 points: perfect                        [ ] 3 points: clearly wrong, some correct elements
[ ] 5 points: essentially correct but with small mistakes [ ] 0 points: for anything else

A test case is a possible input together with the corresponding correct output.

Give two distinct test cases: [ 12 points total, 6 points each ]

[ ] 6 points: perfect                        [ ] 2 points: clearly wrong, some correct elements
[ ] 4 points: essentially correct but with small mistakes [ ] 0 points: for anything else

Test case #1:

\textbf{Input: 21, Output: 12} \hspace{5cm} \textit{This is one way to write a test case.}

Test case #2:

\texttt{postage(41) \rightarrow 53} \hspace{5cm} \textit{This is another. Either/both are good.}
Module 2 - Question 1 [20 points, 4 points each]
The code given below is correct: it compiles without errors. I have added some extra spacing to make this question easier to answer.

Circle, and identify by number, one and only one example of each of the following items in the code below. If you believe no example exists, write "no example" next to that item in the list. To show you how I want the question answered, the first one is done for you.

0. numeric literal
1. conditional statement (entire statement)
2. looping statement (entire statement)
3. empty dictionary literal
4. dictionary lookup
5. variable whose value is a List

```python
def overspent(budget):
    over = {}
    months = list(budget.keys())
    months.pop(0)
    for key in months:
        data = budget[key]
        diff = int(data[0]) - int(data[1])
        if (diff < 0):
            over[key] = diff
    return over
```

Module 2 - Question 2  [20 points total]

Consider a Python function named `sumOfNumbersInList` which has one parameter. The parameter is a List of numbers. The function calculates and returns the sum of all the numbers in the List.

Part 1 [1 point]
What value does `sumOfNumbersInList([ 17, 3, -5 ])` return?

15

Part 2 [1 point]
What value does `sumOfNumbersInList([ 0 ])` return?

0

Part 3 [1 point]
What value does `sumOfNumbersInList([])` return?

0

Part 4 [17 points]

[ ] 17 points: perfect
[ ] 6 points: clearly wrong, some correct elements
[ ] 12 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Define, in Python, the function `sumOfNumbersInList`:

(Many solutions are possible, this is just one example of code you could write. The important thing is whatever code you write, it correctly solves the problem.)

```python
def sumOfNumbersInList(lst) :
    retVal = 0
    for num in lst :
        retVal = retVal + num
    return retVal
```
Module 2 - Question 3  [20 points total]

[ ] 20 points: perfect  [ ] 7 points: clearly wrong, some correct elements
[ ] 14 points: essentially correct but with small mistakes  [ ] 0 points: for anything else

Define a JavaScript function named `printArray` with two parameter. Assume the function will be called with an array of `Strings` and an integer. Define the function so it prints (using `console.log`) each `String` in the array whose length is greater than or equal to the integer.

For example,

```javascript
printArray(["sue", "amy", "bob"], 6)
```

must print nothing, whereas

```javascript
printArray(["foo", "fluffy", "pi", "cake", "rho"], 4)
```

must print the following:

```
fluffy
cake
```

Write your answer below:

*(Many solutions are possible, this is just one example of code you could write. The important thing is whatever code you write, it correctly solves the problem.)*

```javascript
function printArray(rA, minLen) {
    for (var str of rA) {
        if (str.length >= minLen) {
            console.log(str);
        }
    }
}
```
Module 2 - Question 4 [20 points total]

[ ] 20 points: perfect
[ ] 7 points: clearly wrong, some correct elements
[ ] 14 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Study the following code:

```python
import csv

def secrets(filename):
    with open(filename, "r", newline='') as f:
        reader = csv.reader(f)
        for line in reader:
            print(line[1])
    print("Done")

secrets("f.csv")
```

Give possible contents for the file `f.csv` which would cause the above to print:

2
5
My name is Brak
Done

Give your answer below:

(This is just one possible answer, all that matters is that it have 3 rows and the second column contain 2, 5, and My name is Brak).

1, 2, 5
99, 5, 6
10, My name is Brak, 12
Module 3 - Question 1  [20 points, 4 points each]

For each item labelled (a) through (e), match it with its best description from items (1) through (9). Note that four of the items from the numbered list will NOT be used.

(a) AJAX GET request  1) a way to uniquely identify an HTML element
(b) AJAX POST request  2) a way for a client to request code from a server
(c) value attribute  3) a way to get/set the content of an HTML input element
(d) <SCRIPT> tag  4) a way for a client to send data to a server
(e) route  5) a way to specify the specific path(s) that a function on the server will handle

6) a way to encode data so that it can be understood by both the client and server
7) a way for a client to request data from a server
8) a way to get/set the contents of an HTML element that does not handle user input
9) a way to specify key-value pairs as part of a URI

Give your matches below:

(a) best matches with ______ _7_

(b) best matches with ______ _4_

(c) best matches with ______ _3_

(d) best matches with ______ _2_

(e) best matches with ______ _5_
Module 3 - Question 2  [20 points total]

Define a JavaScript function named `activityScore` with one parameter. Assume the function will be called with an array of `Numbers`. Define the function so that it creates an Object. This object should have a key equal to "score" its value should be set to the sum of the array’s entries. Return the JSON blob encoding the Object.

Part 1 [1 point]
What does `activityScore([17, 0, 2])` return?

```
{ "score": 19 }
```

Part 2 [1 point]
What does `activityScore([0])` return?

```
{ "score": 0 }
```

Part 3 [1 point]
What does `activityScore([2, 3, 4, 2, 3, 4])` return?

```
{ "score": 18 }
```

Part 4 [17 points]

[ ] 17 points: perfect
[ ] 6 points: clearly wrong, some correct elements
[ ] 12 points: essentially correct but with small mistakes [ ] 0 points: for anything else

Define, in JavaScript, the function `activityScore`:

```javascript
function activityScore(arr) {
    var sum = 0;
    for (var val of arr) {
        sum = sum + val;
    }
    obj = { "score": sum };  
    return JSON.stringify(obj);
}
```
Module 3 - Question 3  [20 points total]

Part 1 [10 points]
[ ] 10 points: perfect
[ ] 10 points: perfect
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Define a Python function named `serveHTML`. This function should include any annotations needed so that it responds to requests for `/m3q31`. To handle these requests, `serveHTML` should return the contents of the file named `answer.html`. You can assume the file exists and is in the same directory as the file containing this code. Write your answer below:

```python
import bottle

@bottle.route("/m3q31")
def serveHTML() :
    return bottle.static_file("answer.html", root=".")
```

Part 2 [10 points]
[ ] 10 points: perfect
[ ] 7 points: essentially correct but with small mistakes
[ ] 0 points: for anything else

Define a Python function named `loginData`. `loginData` will have a single parameter. This parameter will be a JSON blob. Your function should convert the JSON blob into usable Python data. If both "username" and "password" are keys in that data, `loginData` should return True. If the data is missing at least one of these keys, `loginData` should return False. Write your answer below:

```python
import json

def loginData(jsonBlob) :
    dic = json.loads(jsonBlob)
    retVal = "username" in dic and "password" in dic
    return retVal
```

(Many solutions are possible, this is just one example of code you could write. The important thing is whatever code you write, it correctly solves the problem.)
Module 3 - Question 4 [20 points total]

[ ] 20 points: perfect  [ ] 0 points: for anything else
[ ] 14 points: essentially correct but with small mistakes  [ ] 7 points: clearly wrong, some correct elements

Study this abbreviated HTML and JavaScript code:

```html
<html><head>
...contents unimportant to question and so skipped...
</head>
<body onload="getMessage();">
  Today's Important Message is: <div id="disp"></div><br/><br/>
  How do you respond: <input type="text" id="secretResponse"/>
</body>

function getMessage() {
  ajaxPostRequest("/message", "notVerySecret", displayMessage);
}

function ajaxPostRequest(path, data, callback) {
  var req = new XMLHttpRequest();
  req.onreadystatechange = function() {
    if ((this.readyState === 4) && (this.status === 200)) {
      callback(this.response);
    }
  };
  req.open("POST", path);
  req.send(data);
}

// YOUR CODE WILL GO HERE

Complete the `displayMessage` function. The input to this function will be a JSON blob. Your function should first convert the parameter to usable JavaScript data. You can assume that JavaScript data will be an Object. Get the value associated with the "ans" key from that Object. Get the element, so that it displays the value you found in the Object.

```javascript
function displayMessage(response) {
  obj = JSON.parse(response);
  dat = obj["ans"];  
  divElem = document.getElementById("disp");
  divElem.innerHTML = dat;
}
```
Module 4 - Question 1  [20 points total]

[ ] 20 points: perfect  [ ] 7 points: clearly wrong, some correct elements
[ ] 14 points: essentially correct but with small mistakes  [ ] 0 points: for anything else

Define a Python function named `orderByGoals`. The function has a single parameter. The parameter will be a list of dictionaries. Each dictionary in the list has three keys: "games", "goals", and "assists". The value associated with each key is an `int`. Once your function completes, the list’s entries should be ordered from the dictionary with the fewest goals to the dictionary with the most goals. To accomplish this sort, you should first write the key generator function the `sort` function will need.

```python
def keyGen(dic) :
    return dic["goals"]

def orderByGoals(lis) :
    lis.sort(key = keyGen)
```
Module 4 - Question 2 [20 points total]

[ ] 20 points: perfect  
[ ] 14 points: essentially correct but with small mistakes  
[ ] 7 points: clearly wrong, some correct elements  
[ ] 0 points: for anything else

Define a JavaScript function named `orderByAssists`. The function has a single parameter. The parameter will be an array of Objects. Each Object in the array has three keys: "games", "goals", and "assists". The value associated with each key is a `Number`. Once your function completes, the array’s entries should be ordered from the Object with the fewest assists to the Object with the most assists. To accomplish this sort, you should first write the comparator function the `sort` function will need.

```javascript
function comparatorFn(a, b) {
    if (a["assists"] > b["assists"]) {
        return 42;
    } else if (a["assists"] == b["assists"]) {
        return 0;
    } else {
        return -1;
    }
}

function orderByAssists(rA) {
    rA.sort(comparatorFn);
}
```
Module 4 - Question 3  [20 points total]

Part 1 [10 points]
[ ] 10 points: perfect  [ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes [ ] 0 points: for anything else

What is an HTML injection attack and how do you guard against it?

An HTML injection attack occurs when a malicious user includes HTML code in their input. When this input is then displayed, the HTML will be rendered by the browser and any links followed and any JavaScript is executed. We can guard against this by making certain we encode any inputs. This replaces any special characters with the text that causes HTML to actually render them and so prevents the browser from recognizing any HTML.

Part 2 [10 points]
[ ] 10 points: perfect  [ ] 3 points: clearly wrong, some correct elements
[ ] 7 points: essentially correct but with small mistakes [ ] 0 points: for anything else

Briefly explain how the public key and private key are used in public key encryption. Then explain how this addresses a weakness in shared key cryptography.

Shared key cryptography uses the same key to encrypt AND decrypt data, but this create the problem of securely distributing that key. Public key cryptography solves this problem by using one key for encryption (the public key which is made freely available) and a completely different key for decryption (the private key which is NEVER shared).
Module 4 - Question 4  [20 points total]

Part 1 [10 points]
[ ] 10 points: perfect  [ ] 3 points: clearly wrong, some correct elements  [ ] 7 points: essentially correct but with small mistakes  [ ] 0 points: for anything else

Explain when and why it is better to use:
```python
cursor.execute("SELECT * FROM tableName WHERE field = ?", (variable,))
```
rather than the simpler:
```python
cursor.execute("SELECT * FROM tableName WHERE field = " + variable)
```

The simpler SELECT statement includes variable’s value as part of the code the SQL server executes. If that value came from a user, they could write it so the query matches every row (or every row that they want) and not just those whose value of field matches. The initial statement prevents this attack by separating the code being executed from the data. This should be used any time variable’s value came from an untrusted source (like a user).

Part 2 [10 points]
[ ] 10 points: perfect  [ ] 3 points: clearly wrong, some correct elements  [ ] 7 points: essentially correct but with small mistakes  [ ] 0 points: for anything else

Briefly explain what the `commit()` function does and why it is important.

In order to improve performance, databases often wait before saving any changes back to the database file. This can lead to data being lost if the program ends or database is closed before that saving occurs. The `commit()` function prevents this by directing the database to save any changes right then. This ensures that the database changes are saved and safe, no matter what.