EXAMINATION INSTRUCTIONS

This examination has 9 pages. Check that you have a complete paper. Each candidate should be prepared to produce, upon request, his or her SUNY/UB card.

This examination has 8 questions. Answer all questions. You have 60 minutes 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 must not be taken from the examination room.
► Show all of your work in arriving at an answer, unless instructed otherwise. Partial credit will be awarded as appropriate.
► Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.
► CAUTION – Candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  ♦ Making use of any books, papers or memoranda, calculators or computers, audio or visual cassette players, or other memory aid devices, other than those explicitly authorized by the examiners.
  ♦ Speaking or communicating with other candidates.
  ♦ Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

----------------------------------- DO NOT WRITE BELOW THIS LINE! -----------------------------------

<table>
<thead>
<tr>
<th>MODULE 1</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Subtotal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/20</td>
<td>/20</td>
<td>/20</td>
<td>/20</td>
<td>/80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODULE 2</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Subtotal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/20</td>
<td>/20</td>
<td>/20</td>
<td>/20</td>
<td>/80</td>
</tr>
</tbody>
</table>

/160
Module 1 - Question 1 [20 points, 4 points each]
The code given below is syntactically correct.

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. parameter name
- 2. a string literal
- 3. assignment statement
- 4. a function call
- 5. an argument list

import re
def countWords ( filename ):
    count = {}
    with open ( filename ) as f:
        for line in f:
            wordList = re.split ( "[^a-zA-Z]+" , line )
            for word in wordList:
                if word in count:
                    count [ word ] = count [ word ] + 1
                else:
                    count [ word ] = 1
    return count

There is one other example:
re.split ( ... )

' ( filename ) ' in open call is also an argument list.
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 whatDoesThisDo(x) {
    result = "FOO";
    if (x > 70) {
        return "foo";
    }
    if (x < 50) {
        result = result + "bar";
    }
    else if (x < 30) {
        result = result + "foo";
    }
    if (x < 35) {
        result = result + "-";
    }
    else if (x > 45) {
        result = result + "+";
    }
    else {
        result = result + "*";
    }
    return result;
}
```

[10 points] What does the following statement print?
```javascript
console.log(whatDoesThisDo(65));
```
Write your answer below:

**FOO+**

[10 points] Give a value for x so that when whatDoesThisDo(x) is called the function returns "FOObar*". Write your answer below:

**37 (any value from 35 to 45, inclusive, is acceptable)**
Module 1 - Question 3 [20 points total]

Read this description of how to compute the score on a fictional test:

*The exam consists of 100 questions. The minimum score for each question is zero, and the maximum score is 2. The possible raw score range is therefore zero to 200. However, the score will be reported as a number between 200 and 800. The conversion happens in two steps. First, the raw score is scaled to cover a range of 600 points by multiplying the raw score by 3. Next, the resulting scaled score is translated to the range 200 to 800 by adding 200 to the scaled score.*

Define a Python function named ‘reportedScore’ which converts a raw test score to a scaled and translated test score, as explained above. The function must accept a test score in raw form and return the corresponding scaled and translated score.

Write your answer below:

```
def reportedScore(x):
    return x * 3 + 200
```
Module 1 - Question 4 [20 points total]

Consider a function named ‘letterGrade’ which will be called with one argument (you can assume that it will be called only with an integer value representing a course grade) that computes a string value representing the corresponding letter grade, according to this table:

<table>
<thead>
<tr>
<th>Course grade (%)</th>
<th>Course grade (letter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;= 90</td>
<td>A</td>
</tr>
<tr>
<td>&gt;= 80 and &lt; 90</td>
<td>B</td>
</tr>
<tr>
<td>&gt;= 70 and &lt; 80</td>
<td>C</td>
</tr>
<tr>
<td>&gt;= 60 and &lt; 70</td>
<td>D</td>
</tr>
<tr>
<td>&lt; 60</td>
<td>F</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 set of possible inputs 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:

**Input: 65, output: "D"**

Test case #2:

**Input: -34, output "F"**
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. an empty dictionary literal  
4. the body of a for statement  
5. a dictionary lookup

import re

def countWords( filename ):
    count = {}
    with open( filename ) as f:
        for line in f:
            wordList = re.split( r"[^a-zA-Z]+" , line )
            for word in wordList:
                if word in count:
                    count[ word ] = count[ word ] + 1
                else:
                    count[ word ] = 1
    return count
Module 2 - Question 2 [20 points total]

Consider a Python function named ‘productOfPositivesInList’ which has one parameter. The function will be called with a list of integers and will return the product of all the positive integers in the list (an integer is positive if it is strictly greater than zero). The smallest possible result for any input list is 1.

Part 1 [1 point]
What value does \textit{productOfPositivesInList}([0]) return?

\textbf{1}

Part 2 [1 point]
What value does \textit{productOfPositivesInList}([7, 3, -5]) return?

\textbf{21}

Part 3 [1 point]
What value does \textit{productOfPositivesInList}([ ]) return?

\textbf{1}

Part 4 [17 points]

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

Define, in Python, the function \textit{productOfPositivesInList}:

\begin{verbatim}
def productOfPositivesInList(x):
    answer = 1
    for v in x:
        if v > 0:
            answer = answer * v
    return answer
\end{verbatim}
Module 2 - Question 3 [20 points total]

Define a JavaScript function named makeArray with two parameters. Assume the function will be called with an array of strings and an integer. Define the function so it returns a new array containing all the strings from the original array whose lengths are less than the integer. The original array must not be modified.

For example,

```javascript
makeArray(["sue", "amy", "bob"], 4)
```

must return a new array like this: ["sue", "amy", "bob"], whereas

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

must return a new array like this:

["foo", "pi", "rho"]

Write your answer below:

```javascript
function makeArray(x,y) {
    var result = [];
    for (i in x) {
        if (x[i].length < y) {
            result.push(x[i]);
        }
    }
    return result;
}
```
Module 2 - Question 4 [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

Study the following code:

```python
import csv

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

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

10 :: 2
16 :: 5
93 :: 8
Done

Give your answer below:

Here's one possible solution. There are many others:

0, 10, 2
0, 16, 5
0, 93, 8