Simon Fraser University
Computing Science 120
Midterm Examination
Summer 2017

Time: 45 minutes

Last/Family Name (please, print): 

First Name (please, print): 

Student Number: 

Signature: 

Instructor: Anne Lavergne

This examination has 6 pages.
Let us verify that we have a complete paper.

Instructions:

✓ One cheat sheet allowed. This cheat sheet must …
  • be 8 ½ by 11 inches (letter size) and one-sided,
  • be handwritten,
  • have our name and student number
    on its top right corner, and
  • be handed in with this midterm examination.

✓ No books, calculators, computers, cell phones, or
other materials may be used.

✓ Read each question carefully before answering it.

✓ We must use Python 3.

✓ Let us always comment our code and use the Good
  Programming Style (GPS) discussed in class and
  used in our labs and assignments.

✓ We can assume that all Python code fragments given
  in this exam are syntactically correct, unless stated.

✓ The marks for each question are given in [ ].
  Let us use this to manage our time:
  • 1 mark correspond to 1 minute of work.

Good luck!
1. [14 marks in total – See individual question for marks - No part marks] Write the output each of the following Python code fragments produce (on the computer monitor screen) in the box to its right. If we encounter an error, we do not have to fix it. All we have to do is write “error” in the box to the right.

<table>
<thead>
<tr>
<th>Python Code Fragment</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) [2 marks] numbers = list(range(0,9)) aSlice = numbers[0::2] print(aSlice)</td>
<td></td>
</tr>
<tr>
<td>b) [4 marks] Name = &quot;Xiao&quot; name = &quot;Xiao&quot; taxable = 45900.5124 rate = 15 age = 26 record = &quot;Name: %s\nage: %d &quot;%(name, age) record += &quot;\ntaxable: $%0.2f&quot;%taxable record += &quot;\nrate: %d%%&quot;%rate print(&quot;{}&quot;.format(record))</td>
<td></td>
</tr>
<tr>
<td>c) [2 marks] grades = [1, 2, 3, 4] grades.remove(2) print(grades)</td>
<td></td>
</tr>
<tr>
<td>d) [2 marks] print(str(16 - 6 // 2 * 3) * 3 )</td>
<td></td>
</tr>
</tbody>
</table>
| e) [4 marks] symbol = '&' stop = 6 aVar = list( ) for number in range(1, stop+1) :
  aChar = str(number)
  if aChar in "13579" :
    aVar.append(symbol)
  else:
    aVar.append(aChar)
print("%s" % aVar) | |
2. [12 marks] Consider the following Python code fragment:

```python
grade = int(input("Please, enter a grade between 0 and 100 : "))
if grade < 0:
    print("Negative grade.")
elif grade > 100:
    print("Grade > 100.")
elif 0 <= grade <= 49:
    print("F")
elif 50 <= grade <= 69:
    print("C")
elif 70 <= grade <= 89:
    print("B")
ext:
    print("A")
```

In the table below, create the minimum number of test cases we would need to completely test this Python code fragment.

<table>
<thead>
<tr>
<th>Test Case #</th>
<th>Test Data</th>
<th>Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. [19 marks] **Problem Statement**: Write a complete Python program that, given a list of string elements, figures out how many of these string elements contain words (i.e., contain only letters) and how many of these string elements contain numbers (i.e., contain only digits).

To exemplify the behaviour and the output of this Python program, here are some sample runs:

1. If `aList = ['apple', 'banana', 'pear']`, then our Python program produces:
   
   ```
   ['apple', 'banana', 'pear'] contains 3 word(s) and 0 number(s)
   ```

2. If `aList = ['me', '2', 'you', '3', 'her', 'them', '78']`, then our Python program produces:
   
   ```
   ['me', '2', 'you', '3', 'her', 'them', '78'] contains 4 word(s) and 3 number(s)
   ```

3. If `aList = []`, then our Python program produces:
   
   ```
   [] contains 0 word(s) and 0 number(s)
   ```

4. If `aList = ['23', '24', '25', '26']`, then our Python program produces:
   
   ```
   ['23', '24', '25', '26'] contains 0 word(s) and 4 number(s)
   ```

5. If `aList = ['sun']`, then our Python program produces:
   
   ```
   ['sun'] contains 1 word(s) and 0 number(s)
   ```

6. If `aList = ['9']`, then our Python program produces:
   
   ```
   ['9'] contains 0 word(s) and 1 number(s)
   ```

We do not have to create function(s) in our Python program.

We must not ask the user to enter a list. Instead, we can simply assign a list to our list variable in our program as follows:

```python
aList = ['apple', 'banana', 'pear']
```

Words containing a mixture of letters, digits and/or symbols such as '2s*u$n' or '5 cats' cannot/will not be used in our Python program.

If we find it useful to start by designing an algorithm, let’s feel free to do so.

Let’s use the next page to answer this question.