CSC 120 Midterm
Instructor: Katie Fraser
October 27, 2016

Last name: First name:

UTORid: Student number:

Do not turn this page until you have received the signal to start.
(Please fill out the identification section above, write your name on the back of the test, and read the instructions below.)

Good Luck!

• You have 50 minutes to write the test.
• You are not permitted any aids, notes, books, or calculators.
• This test is out of 25 points.
• This booklet is double-sided. There are eight pages in total.
• Comments are not required except where indicated, although they may help us mark your answers.
• If you use any space for rough work, indicate clearly what you want marked.
• Do not remove any pages from the exam booklet.
• You may use a pencil; however, work written in pencil will not be considered for remarking.
• As always, you are expected to adhere to the U of T student code of academic conduct.

<table>
<thead>
<tr>
<th>Grade earned</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>25</td>
</tr>
</tbody>
</table>
Question 1 (4 marks)

Implement the function described in this Python docstring.

```python
def upper_all(str_list):
    """(list of str) -> NoneType
    Modify str_list, a list of strings, so that every string is converted to uppercase.
    
    >>> test_list = ['a','bob','Hello']
    >>> upper_all(test_list)
    >>> test_list
    ['A', 'BOB', 'HELLO']
    """
```
Question 2 (6 marks)

Write a function that returns a copy of a string with all the vowels (both uppercase and lowercase) removed. For example, if the input is:

'Oh boy, I love programming!'

then the output should be:

'h by, lv prgrmmng!'

Write a docstring for your function that includes two examples (different from the one given above). For the purposes of this question, the vowels will be a, e, i, o, and u.
Question 3 (4 marks)

Consider the following two function definitions (docstrings excluded due to space). Beside each code fragment in the table below, write what is printed when the code fragment is executed.

```python
def mystery(num):
    total = 0
    if num > 10:
        total = total + 10
    elif num < 20:
        total = total + 20
    else:
        total = total + 1
    return total

def enigma(num):
    total = 0
    if num > 10:
        total = total + 10
    if num < 20:
        total = total + 20
    else:
        total = total + 1
    return total
```

<table>
<thead>
<tr>
<th>Code</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>print(mystery(15))</td>
<td>31</td>
</tr>
<tr>
<td>print(mystery(30))</td>
<td>31</td>
</tr>
<tr>
<td>print(enigma(15))</td>
<td>31</td>
</tr>
<tr>
<td>print(enigma(30))</td>
<td>31</td>
</tr>
</tbody>
</table>
Question 4 (6 marks)

Assume L is a list of lists, where each nested list corresponds to a different student and contains that student’s grades for their assignments. Write a function that asks the user which assignment they would like to retrieve, then return a list containing all of the students’ grades for that assignment. You can assume the grade for Assignment 0 is stored at index 0 of the nested lists, the grade for Assignment 1 is stored at index 1 of the nested lists, and so on. All the nested lists have the same length. If the user asks for an Assignment that does not exist, return an empty list. You can assume the user will enter a non-negative integer. A docstring is given.

```python
def ask_for_assignment_grades(L):
    """ (list of list of number) -> list of number
    
    L is a list of lists, where each nested list contains a student's grades for their assignments. Prompt the user for an assignment number, then return a list of grades for the requested assignment. If the requested assignment does not exist, return an empty list.
    
    >>> ask_for_assignment_grades([[90, 95], [85, 88], [75, 90]])
    Which assignment would you like? 0
    [90, 85, 75]
    >>> ask_for_assignment_grades([[90, 95], [85, 88], [75, 90]])
    Which assignment would you like? 2
    []
    """
```

```
Question 5 (5 marks)

Here are several short chunks of Python code. Beside each chunk, write the output it will produce. If an error would occur, write ERROR and a short explanation.

```
x = 5
y = 1
x = y
y = 2
print(x)
```

```
x = 2 + 10 % 3
print(x)
```

```
s = 'mississippi'
print(s[:5])
```

```
s = 'csc' + 120 * 3
print(s)
```

```
for i in [True, False]:
    for j in [True, False]:
        print(i or j)
```
[Use the space below for rough work. This page will not be marked, unless you clearly indicate the part of your work that you want us to mark.]
Short Python function/method descriptions:

```python
__builtins__:
    int(x) -> int
        Convert x to an integer, if possible. A floating point argument will be truncated towards zero.
    len(x) -> int
        Return the length of list, tuple, or string x.
    print(value) -> NoneType
        Prints the values.
    input([prompt]) -> str
        Read a string from standard input. The trailing newline is stripped. The prompt string, if given, is printed without a trailing newline before reading.
    range([start], stop, [step]) -> list-like-object of int
        Return the integers starting with start and ending with stop - 1 with step specifying the amount to increment (or decrement). If start is not specified, the sequence starts at 0. If step is not specified, the values are incremented by 1.
    str(x) -> str
        Return an object converted to its string representation, if possible.

str:
    x in s --> bool
        Produce True if and only if x is in s.
    S.count(sub[, start[, end]]) -> int
        Return the number of non-overlapping occurrences of substring sub in string S[start:end]. Optional arguments start and end are interpreted as in slice notation.
    S.find(sub[,i]) -> int
        Return the lowest index in S (starting at S[i], if i is given) where the string sub is found or -1 if sub does not occur in S.
    S.isalpha() -> bool
        Return True if and only if all characters in S are alphabetic and there is at least one character in S.
    S.isalnum() -> bool
        Return True if and only if all characters in S are alphanumeric and there is at least one character is S.
    S.isdigit() -> bool
        Return True if and only if all characters in S are digits and there is at least one character in S.
    S.islower() -> bool
        Return True if and only if all cased characters in S are lowercase and there is at least one cased character in S.
    S.isupper() -> bool
        Return True if and only if all cased characters in S are uppercase and there is at least one cased character in S.
    S.lower() -> str
        Return a copy of the string S converted to lowercase.
    S.replace(old, new) -> str
        Return a copy of string S with all occurrences of the string old replaced with the string new.
    S.upper() -> str
        Return a copy of the string S converted to uppercase.

list:
    x in L --> bool
        Produce True if and only if x is in list L
    L.append(object) -> NoneType
        Append object to end of list L.
    L.extend(iterable) -> NoneType
        Extend list L by appending elements from the iterable. Strings and lists are iterables whose elements are characters and list items respectively.
```