CSC 108H1 F 2011 Test 2
Duration — 45 minutes
Aids allowed: none

Student Number: _____

Lab day, time, room: _____

Last Name: _____ **First Name:** _____

Lecture Section: L5101 **Instructor: Daniel Zingaro**

---

*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!

This midterm consists of 3 questions on 8 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* Comments are not required except where indicated, although they may help us mark your answers. They may also get you part marks if you can't figure out how to write the code.
If you use any space for rough work, indicate clearly what you want marked.

1: _____/ 7
2: _____/ 6
3: _____/ 4

TOTAL: _____/17

Question 1. [7 MARKS]

The greatest video game music composer of all time is Motoi Sakuraba. Unfortunately, he has become mixed up with a bunch of other data in the Python dictionaries below. For each of the three subquestions on this page, add one or more lines of code to print `Sakuraba` **on a single line** by extracting the proper part of the dictionary. Each subquestion is independent.

Part (a) [1 MARK]

```
composers = {'a': 'Kondo', 'b': 'Sakuraba', 'c': 'Kikuta'}
```

Part (b) [1 MARK]

```
composers = {'a': {'Sakuraba': 'b'}}
```

Part (c) [2 MARKS]

You are required to use a loop in this one.

```
composers = {1:'S', 2:'a', 3:'k', 4:'u',
             5:'r', 6:'a', 7:'b', 8:'a'}
```

Part (d) [1 MARK]

Assume that I try to invert the following dictionary, using inversion functions we have written in lecture:

```
d = {'first': [1,2,3]}
```

Can this dictionary be inverted? Explain in one or two sentences what will happen.

Part (e) [2 MARKS]

In the box beside the code below, write its output. If it would generate an error, say so, and give the reason for the error.

```
L1 = [[1, 2], [3, 4]]
L2 = L1[:]
L1[1].append (99)
print L1
print L2
```

Question 2. [6 MARKS]

A **composer** file consists of zero or more **composer blocks**. Each composer block consists of the following lines, in order:

- A line containing the composer's name
- Zero or more lines, each naming one song written by the composer

Each composer, except for the last, is followed by two lines of five asterisks each. Here is an example:

```
Uematsu
Silent Light
Sealed Door
*****
*****
Sasai
*****
*****
Kawasaki
Longing for the Past
Guardians
Unexplored Road
```

In this sample file, Uematsu has 2 songs; Sasai has 0 songs; and Kawasaki has 3 songs. Overall, there are three composers, five total songs; and the composer with the most number of songs is Kawasaki.

(See next page for question. You may use the area below for rough work, but it will not be marked unless you clearly indicate the part of it you want us to mark.)

Write the following function according to its docstring.

```
def composer_info (f):
  '''f is an open composer file. Return a list of three elements:  the first element is the number
  of composers in the file; the second is the total number of songs; and the third is the name
  of the composer (an str) with the most songs in the file (if multiple composers have the most
  songs, return any one of them).'''
```

Question 3. [4 MARKS]

Write the following function according to its docstring.

```
def indices_and_elements (L):
  '''Return a list of tuples, one for each element in list L, in order. Each
  tuple should contain the index of the element from L, and the element
  itself. For example, indices_and_elements (['this', 'is', 'fun']) should
  return [(0, 'this'), (1, 'is'), (2, 'fun')]'''
```

[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.]

Last Name: _____ **First Name:** _____

Short Python function/method descriptions:

```
  len(x) -> integer
    Return the length of the list or string x.
  sum(x) -> integer
    Return the sum of the elements in the list x.
  open(name[, mode]) -> file object
    Open a file.
  range([start], stop, [step]) -> list of integers
    Return a list containing the integers starting with start and ending with
    stop - 1 with step specifying the amount to increment (or decrement).
dict:
  D[k] -> value
    Return the value associated with the key k in D.
  k in d -> boolean
    Return True if k is a key in D and False otherwise.
  D.keys() -> list of keys
    Return the keys of D.
  D.values() -> list of values
    Return the values associated with the keys of D.
  D.items() -> list of 2-tuples.
    Return a list of D's (key, value) pairs.
file (also called a "reader"):
  F.close()
    Close the file.
  F.read([size]) -> string
    Read at most size bytes; with no size, read until EOF.
  F.readline([size]) -> string
    Read next line, retaining newline; return empty string at EOF.
str:
  S.find(sub[,i]) -> integer
    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.replace(old, new) -> string
    Return a copy of string S with all occurrences of the string old replaced
    with the string new.
  S.split([sep]) -> list of strings
    Return a list of the words in S, using string sep as the separator and
    any whitespace string if sep is not specified.
  S.startswith(prefix) -> boolean
    Return True if S starts with the specified prefix and False otherwise.
  S.strip() --> string
    Return a copy of S with leading and trailing whitespace removed.
list:
  L.append(x)
    Append x to the end of the list L.
  L.index(value) -> integer
    Return the lowest index of value in L.
  L.insert(index, x)
    Insert x at position index.
```