

CSC 108H1 F 2011 Test 1
Duration — 45 minutes
Aids allowed: none

Student Number: _____

Last Name: _____ First Name: _____

Lecture Section: L0301

Instructors: Craig

*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 6 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* Comments and docstrings 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. No error checking is required: assume all user input and all argument values are valid.

1: _____/ 7

2: _____/ 8

3: _____/ 5

If you use any space for rough work, indicate clearly what you want marked.

TOTAL: _____/20

Question 1. [7 MARKS]

In each question below, fill in the box with python code that will make the program behaviour match the comments. You may **not** make any other changes to the code. If there is nothing that you could write in the box to make the program work as specified, put a big **X** through the box.

Part (a) [1 MARK]

```
answer = ''
```

```
while
```

```
    answer = raw_input('Leafs or Habs? ')
```

```
# If we get to this line, answer is either Leafs or Habs.
```

Part (b) [1 MARK]

```
s = "big old cat"
```

```
# Replace 'old' with 'fat' so that s becomes 'big fat cat'
```

```
s[4:7] =
```

Part (c) [1 MARK]

Assume `funny` and `slob` are boolean variables indicating characteristics of your room-mate.

```
# Set boolean variable happy to True if your room-mate is funny but isn't a slob,  
# and False otherwise.
```

Part (d) [1 MARK]

```
name = raw_input("Enter your name: ")
```

```
# If name is longer than 20 characters, shorten it to only the first 20 characters.
```

```
if
```

```
    name = name[:20]
```

Part (e) [1 MARK]

```
def is_char_at(s, ch, x):
    '''Return True iff string s has character ch at index x.'''
    return s[x] == ch

course = raw_input("What is your favourite course? ")
# Use the function is_char_at to check if the course starts with 'C' and print a
# message if it does.
```

```
if  :

    print 'Your favourite could be CSC 108!'
```

Part (f) [1 MARK]

```
# Students have finished 3 labs so far and each lab grade is either 0 or 1.
```

```
lab1 = int(raw_input('Please enter your lab1 grade: '))
lab2 = int(raw_input('Please enter your lab2 grade: '))
lab3 = int(raw_input('Please enter your lab3 grade: '))
```

```
# Calculate and print the average lab mark for this student.
```

```
print 
```

Part (g) [1 MARK]

```
def raise_to_A(grade):
```

```
 :
```

```
mark = 79.4
# Raise this student's mark to 80.0.
raise_to_A(mark)
# Print the updated mark.
print mark
```

Question 2. [8 MARKS]**Part (a)** [4 MARKS]

Complete the following function according to its docstring.

```
def mute_channel(snd, channel):  
    '''String channel is either 'right' or 'left'. Set all  
    the sample values of Sound snd to zero in the channel indicated.'''
```

Part (b) [4 MARKS]

Write a main block that allows the user to choose a file, and asks the user using the prompt “Which channel? (enter left or right):” to specify which channel to mute. It then plays the sound with the appropriate channel muted. Assume that the user chooses a .wav file, and enters a valid answer to your question about which channel.

Make sure that if this module is imported, none of this code executes – just the function definition.

Question 3. [5 MARKS]

Consider this Python program.

```
def mystery(s, c, n):  
  
    result = ''  
    for ch in s:  
        if ch == c:  
            result += c * n  
        else:  
            result += ch  
    return result  
  
if __name__ == '__main__':  
    print mystery('Diane', 'a', 3)  
    print mystery('Michelle', 'e', 2)  
    print mystery('Dan', 'n', 0)  
    print mystery('CSC108 instructors', 'c', 3)
```

Part (a) [2 MARKS]

The output from the program is four lines. Show it in this table.

Part (b) [3 MARKS]

Write a good docstring for the function.

Last Name: _____ First Name: _____

Short Python function/method descriptions:

```
__builtins__:
abs(number) -> number
    Return the absolute value of the given number.
max(a, b, c, ...) -> value
    With two or more arguments, return the largest argument.
min(a, b, c, ...) -> value
    With two or more arguments, return the smallest argument.
raw_input([prompt]) -> string
    Read a string from standard input. The trailing newline is stripped. The prompt string,
    if given, is printed without a trailing newline before reading.
int:
int(x) -> integer
    Convert a string or number to an integer, if possible. A floating point argument
    will be truncated towards zero.
media:
choose_file() --> str
    Prompt user to pick a file. Return the path to that file.
sound:
copy(Sound) --> Sound
    Return a copy of the given Sound.
create_sound(int) --> Sound
    Create a sound with the specified number of samples. All sample values are 0.
get_left(sample) --> int
    Return the left value of the given sample.
get_right(sample) --> int
    Return the right value of the given sample.
load_sound(str) --> Sound
    Return a Sound object from file with the given filename.
set_left(sample, int)
    Set the left value of the given sample to the given int value.
set_right(sample, int)
    Set the right value of the given sample to the given int value.
play(Sound)
    Play the given Sound.
```