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

Student Number: 9 9 9 2 9 2 5 0 9

Last Name: Qiu **First Name:** Rui

Lecture Section: L0101 **Instructor:** **Campbell**

*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 4 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.
If you use any space for rough work, indicate clearly what you want marked.

# 1:	1 / 2
# 2:	1 / 2
# 3:	4 / 8
# 4:	7 / 8
TOTAL:	13 /20

Test 1

CSC 108H1 S

Question 1. [2 MARKS]

Part (a) [1 MARK] What is the output of the following?

```
pic = media.create_picture(50, 100)
pic2 = media.add_text(pic, 0, 0, 'test', media.yellow)
print type(pic2)
```

This function has no return value, so type is 'None Type'

0 ~~str~~ picture

Part (b) [1 MARK] Rewrite the following code without an if-statement.

```
if ketchup and not mustard:
    return True
else:
    return False
```

1 return ketchup and not mustard

Question 2. [2 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.

Part (a) [1 MARK]

```
name = 'Matthew'
age = 3

# Print the following: Matthew is 3!
```

~~str(name + str(is) + age)~~

0 print ~~%s is %d~~ ~~name + 'is' + age name +~~ % (name, age)

~~name + str(is) + str(age)~~

not used

Part (b) [1 MARK]

```
pic = media.load_picture(media.choose_file())

# get the pixel at (10, 4)
```

1 pix=media.get_pixel (pic, 10, 4)

```
# set the pixel at (10, 4) to yellow
media.set_color(pix, media.yellow)
```

Student #: ___________

Question 3. [8 MARKS]

Part (a) [4 MARKS] Complete the following function according to its docstring description.

3

```
def change_green(pic, factor):
    '''(Picture, float) -> Picture
    Return a new picture that is a copy of pic, but with each pixel's green color
    component set to its original value multiplied by factor. factor is a value
    between 0.0 and 1.0, inclusive.'''
    pic=media.load_picture(media.choose_file()) X
    pic2= media.copy(pic)
    ~~factor = float( random <=1.0 and random >= 0.0~~)
    for pixel in pic2:
        green = media.get_green(pixel)
        media.set_green(pixel, ~~green * factor)~~ int(green * factor))
    return pic2
```

-0.5 you are given pic

```
factor = float(~~random value~~) raw-input
~~rad raw-input~~
~~random value~~ <~~=1.0 and random~~
~~random value >= 0.0~~.
raw_input
```

-0.5 you are given factor

1

Part (b) [4 MARKS]

Write a main block that allows the user to choose a file, prompts the user with, `'Enter a value between 0.0 and 1.0, inclusive: '`, applies the change_green function from part (a) to the picture in that file using the value entered by the user, and displays the resulting picture. You may assume that the user chooses a valid picture file and enters a valid value.

```
if __name__ == '__main__':
    pic - media. load_picture (media.choose-file())
    ~~media.show(media change_green(pic, factor))~~
    raw_input = "Enter a value between 0.0 and 1.0" + " "
    factor = float (raw-input ('Enter a value . - ))
    ~~media.show(media change_green(pic,~~ f

    ~~pic2=media~~.
    media.show(change_green(pic, ~~raw_input~~ factor))
```

Question 4. [8 MARKS]

Consider the following two .py files, which are saved in the same directory (folder).

module_a.py:

```
def f(s):
    result = ’’

    for char in s:
        if char == char.upper():
            result = result + char

    return result

if __name__ == ’__main__’:
    print f(’EFg’)

# this code is not inside the
# body of the if-statement
print f(’aBcde’)
```

module_b.py:

```
import module_a

def g(s):
    answer = module_a.f(s)
    return len(answer)

if __name__ == ’__main__’:
    print module_a.f(’WXyZ’)
    print g(’TeSTiNg’)
```

This question continues on the next page. You may use the space below for rough work.

Test 1

Part (a) [1 MARK] 1 ✓

How many lines of output are produced when `module_b` is executed (by clicking Run)?

Circle one: ~~2 lines~~ [✓ 3 lines] 4 lines

Part (b) [4 MARKS] 3

In the table below, show the output from running `module_b`. If there are fewer than four lines of output, leave the unused box(es) empty.

(~~WXZ~~) ~~EF~~ B	← in __main__ : ← print f('aBcde') not in main?	−1
(~~3~~) WXZ ✓		
4 ✓		
✓		

Part (c) [3 MARKS] 2

Write a good docstring for the function `f` from `module_a`.

'''(String) -> String
Return the capitalized letters in the original string ^ to form a new string.'''

S ← call it by name
−1

Last Name: Qiu First Name: Rui

Short Python function/method descriptions:

```
__builtins__:
  len(x) -> int
    Return the length of the list, tuple, dict, or string x.
  raw_input([prompt]) -> str
    Read a string from standard input.  The trailing newline is stripped.
float:
  float(x) -> float
    Convert a string or number to a floating point number, if possible.
int:
  int(x) -> int
    Convert a string or number to an integer, if possible.  A floating point
    argument will be truncated towards zero.
media:
  add_text(pic, x, y, s, col)
    Draw the str s in Color col on Picture pic starting at (x, y).
  choose_file() --> str
    Prompt user to pick a file. Return the path to that file.
  copy(Picture) -> Picture
    Return a copy of the Picture.
  create_picture(int, int) --> Picture
    Given a width and a height, return a Picture with that width and height.  All pixels are white.
  get_blue(Pixel) --> int
    Return the blue value of the given Pixel.
  get_color(Pixel) --> Color
    Return the Color object with the given Pixel's RGB values.
  get_green(Pixel) --> int
    Return the green value of the given Pixel.
  get_pixel(Picture, int, int) --> Pixel
    Given x and y coordinates, return the Pixel at (x, y) in the given Picture.
  get_red(Pixel) --> int
    Return the red value of the given Pixel.
  load_picture(str) --> Picture
    Return a Picture object from file with the given filename.
  set_blue(Pixel, int)
    Set the blue value of the given Pixel to the given int value.
  set_color(Pixel, Color)
    Set the RGB values of the given Pixel to those of the given Color.
  set_green(Pixel, int)
      Set the green value of the given Pixel to the given int value.
  set_red(Pixel, int)
    Set the red value of the given Pixel to the given int value.
  show(Picture)
    Display the given Picture.
  Colors:    black:  RGB: 0, 0, 0     white:  RGB: 255, 255, 255    yellow: RGB: 255, 255, 0
str:
  str(x) -> str
    Convert an object into its string representation, if possible.
  S.upper() -> string
    Return a copy of the string S converted to uppercase.
```

END OF EXAMINATION