Carleton University
SYSC 1005 – Introduction to Software Development – Fall 2017
Sample Midterm Exam - Solutions

Question 1 [8 marks]

To check if your answers are correct, type the expressions in the Python 3.x shell. Each correct answer was worth 1 mark.

Question 2 [4 marks]

The easiest way to do this question is to trace the code, step-by-step, and draw diagrams showing the bindings between variables `image1` and `image2` and parameters `first` and `second`, to the two images. This techniques was presented in one of the sets of slides on image processing. Each correct solution was worth 2 marks.

The following statements, when executed in the Python shell, run without error.

```
>>> image1 = load_image(choose_file())
>>> image2 = load_image(choose_file())
>>> mystery_filter(image2, image1)
>>> show(image1)
```

The picture that was just displayed is

A. **entirely crimson.**

B. entirely slateblue.

C. the original image that was bound to `image1`.

D. the original image that was bound to `image2`.

```
>>> show(image2)
```

The picture that was just displayed is

A. entirely crimson.

B. entirely slateblue.

C. the original image that was bound to `image1`.

D. **the original image that was bound to `image2`.**

Question 3 [5 marks]

To check if your diagrams are correct, type the script in the PyTutor editor and visualize the execution of the code.

Question 4 [2 marks]

When this script was executed, we chose an image in which all the pixels have the same RGB colour, `(60, 120, 150)`. Briefly describe the modified image that was displayed when `show` was called. If no image was displayed, explain why.

Solution:

In the modified image, the only pixel that has changed is the one in the lower-right corner, which is now this shade of gray: (110, 110, 110).

Note: no marks were awarded for explaining why the modified image looks the way it does. The question didn't ask for this. 1 mark was awarded for indicating which pixel was modified. 1 mark was awarded for indicating the colour of the modified pixel.

Question 5

Which of the following are not legal names for Python variables?

A. `fiveDogs`

B. `5Dogs`

C. `five_dogs`

D. `_5_dogs`

E. `five dogs`

Question 6

Which of the following statements about Python functions is correct?

A. The body of a function must contain at least one return statement.

B. Assigning a value to a function parameter changes the corresponding function argument.

C. A function call must contain at least one argument.

D. If a function has a return statement, any call to that function must appear on the right side of an assignment statement.

E. None of the statements are correct.

Question 7

The colour of the pixel is now:

A. `(117, 192, 39)`

B. `(192, 117, 39)`

C. `(117, 39, 192)`

D. `(192, 39, 117)`

E. `(116, 116, 116)`

F. `A different colour than any of the ones listed here.`

Question 8

The length of time for which parameter `a` exists is:

A. Really long.

B. From the time the function is called until the function returns.

C. From the time the function object is created until the script finishes executing.

D. None of the above.

To verify that choice B is correct, type the script in the PyTutor editor and visualize the execution of the code.

Question 9

Which statements bind `0.8` (a value of type `float`) to variable `f`?

A. `f = 8 / 10`

B. `f = 8 // 10`

C. `f = 8.0 / 10.0`

D. `f = 8.0 // 10.0`

E. `f = float(8 // 10)`

F. `f = float(8) // 10`

To verify that A and C are the correct choices, type the five statements in the Python 3.x shell. After each statement is executed, evaluate `f`.

Question 10

Which type contract should be added to this function's docstring?

A. `(int, int) -> float`

B. `(int, int) -> int`

C. `(int, int) -> None`

D. `(float, float) -> float`

E. `(float, float) -> int`

F. `(float, float) -> None`

G. `(number, number) -> float`

H. `(number, number) -> int`

I. `(number, number) -> None`

J. `This function doesn't need a type contract.`

Question 11

What is displayed when `bunny` is called from the shell?

A.
```
29
29
```

B.
```
73
73
```

C.
```
29
73
```

D.
```
73
29
```

E.
```
29
```

F.
```
73
```

To determine why choice D is correct, type the script in the PyTutor editor, add a call to `bunny`, and visualize the execution of the code.

Question 12

The missing code is:

```
A.    temp = a
      b = a
      a = temp

B.    temp = a
      b = a
      b = temp

C.    temp = b
      b = a
      b = temp

D.    temp = a
      a = b
      b = temp

E.    temp = b
      a = b
      b = temp
```

To determine why D is the only correct choice, use PyTutor to visualize the script five times, each time replacing the missing code with one of fragments A through E.

Question 13

When the function returns, how many pixels have had their colour set to yellow?

```
A.    0

B.    1

C.    63

D.    117

E.    7371

F.    7552

G.    256,000
```

One way to verify this is to modify the function to count the number of loop iterations (**changes are boldfaced**) - one pixel is modified per iteration - and call the function from the Python shell:

```
def make_yellow(image):
    yellow = create_color(255, 255, 0)
    counter = 0
    for pixel in image:
        x, y, (r, g, b) = pixel
        print('Modifying pixel @ x =', x, 'y =', y)
        set_color(image, x, y, yellow)
        counter = counter + 1
    print(counter)
```

Question 14

What value(s) are printed by this code?

A. `False`

B. `3, 4`

C. `3, 4, 5`

D. `1, 2, 3, 4, 5`

E. `0, 1, 2, 3, 4`

F. `0, 1, 2, 3, 4, 5`

G. `Nothing is printed.`

To verify this, use PyTutor to visualize the execution of the script.

Question 15

What numbers does function `mystery` print?

A. `10 11`

B. `11 10`

C. `7 14`

D. `14 7`

E. `None; instead, an error message is displayed.`

To verify this, use PyTutor to visualize the execution of the script.

Question 16

What does this code print?

A. `x equals 1`

B. `x equals 2`

C. `not 1 or 2`

D.
```
x equals 1
x equals 2
```

E.
```
x equals 1
not 1 or 2
```

F.
```
x equals 2
not 1 or 2
```

G. `Nothing`

To verify this, use PyTutor to visualize the execution of the script.

Question 17

For what values bound to `x` will `2` be among the values printed?

A. **`x < 0`**

B. `x >= 0`

C. `x < 20`

D. `All values of x`

E. `None of the above`

To verify this, use PyTutor to visualize the execution of the script. Remember to bind `x` to an integer before the `if` statement.

Question 18

Which of the following code fragments is equivalent to this one? "Equivalent" means that both fragments would print the same values.

A.
```
if b < c:
    d = a * 2
else:
    a = 2
print(a, b, c, d)
```

B.
```
if b <= c:
    d = a * 2
else:
    a = 2
print(a, b, c, d)
```

C.
```
if b >= c:
    d = a * 2
else:
    a = 2
print(a, b, c, d)
```

D.
```
if b > c:
    d = a * 2
else:
    a = 2
print(a, b, c, d)
```

To verify this, use PyTutor to visualize the execution of the five scripts.