University of Saskatchewan
Department of Computer Science
CMPT 140
Midterm Exam
October 26, 2017

Marks: 54 **Time: 90 minutes**

Name: _____ **NSID:** _____ **Student #:** _____

Please circle your instructor: *Jeff Long (Section 01)* / *David Kreiser (Section 03)*

********** Solution & Marking Guide **********

Page	1-7	8	9	10	11	**Total**
Marks						
Max	33	6	6	4	5	54

NSID: _____

Multiple Choice (OPSCAN)

Answer the following questions using the OPSCAN sheet provided.

(1) 1. Which of the following is **NOT** true about **actions** in algorithms?

A. the algorithm's recipient can perform the action without further elaboration

B. actions must accomplish something worthwhile or useful

C. actions must be a command to do something

D. the algorithm's recipient must be capable of carrying out the action

E. All of the above are true about actions in algorithms

(1) 2. Which of the following statements about problems and algorithms is **TRUE**?

A. a given algorithm solves only one problem

B. a problem is a task to be carried out and an algorithm is a list of actions to solve a task

C. multiple algorithms may be suitable for solving the same problem

D. All of the above statements are true

E. None of the above statements are true

(1) 3. Which feature(s) of the Python language directly enable(s) effective use of **encapsulation**?

A. libraries

B. conditional branching (if-statements)

C. calling and defining functions

D. Both A and B

E. Both A and C

F. Both B and C

(1) 4. **Refinement** is the process of:

A. adding more details to an algorithm

B. converting a computer program to 1s and 0s so the machine can understand it

C. separating a related group of actions and giving them a name

D. strategically removing details from an algorithm

E. composing an algorithm in plain English pseudocode

(1) 5. On a 200 by 200 Processing canvas, the coordinate (0, 200) can be found:

A. at the top-left corner

B. at the top-right corner

C. at the center

D. at the bottom-left corner

E. at the bottom-right corner

(1) 6. What is drawn on the canvas by the following Processing code?

```
size(300,300)
line(0,0,300,300)
line(0,300,300,0)
```

A. lines forming an "X" through the center of the canvas

B. one rectangle covering the entire canvas

C. one triangle in the center of the canvas

D. lines forming two vertical (up and down) bars across the canvas

E. lines forming two horizontal (side to side) bars across the canvas

For questions 7 and 8, refer to the following Python code:

```
def draw_tree(x,y):
    rect(x-5,y,10,50)
    ellipse(x,y,40,40)
```

(1) 7. In the above code, the symbol `x` is an example of:

A. a string

B. an argument

C. a parameter

D. a control statement

E. a literal

(1) 8. In the above code, the line `rect(x-5,y,10,50)` is an example of:

A. a control statement

B. a function definition

C. a literal

D. a function call

E. a variable

(1) 9. Assume someone else had written a Processing function with the following function header:

```
def divide_by_two(number)
```

From the function header **alone**, you can **definitely** conclude:

A. the function will divide the parameter `number` by two

B. the function must be called with one argument

C. the parameter `number` must be either an integer or a floating point data type

D. You can conclude both A and B

E. You can conclude all of A, B and C

F. None of the above can be concluded from the function header alone

(1) 10. What is printed to the console by the following program?

```
def print_message(msg):
    msg = msg * 2
    print("msg")

print_message("COOKIE!")
```

A. COOKIE!

B. COOKIE!COOKIE!

C. msg

D. nothing is printed (but the program doesn't crash)

E. The program crashes due to an error

(1) 11. In Processing, the triplet (0,255,0) represents which colour?

A. green B. black C. blue D. red E. None of the above

(1) 12. If the triplet (100,0,0) represents a dark red in Processing, which triplet below represents a lighter red?

A. (50,0,0) B. (100,100,100) **C. (200,0,0)** D. (100,50,50) E. (100,200,200)

(1) 13. In Processing, the triplet (255,0,255) represents which colour?

A. white B. green C. red D. blue **E. None of the listed options**

(1) 14. What is drawn on the canvas by the following Processing code?

```
fill(255)
rect(20, 20, 10, 10)
fill(0,0,0)
rect(20, 40, 10, 10)
rect(20, 60, 10, 10)
```

A. three overlapping gray squares

B. one white square and two black squares with no overlap

C. one black square and two white squares with no overlap

D. a single white square

E. None of the above

(1) 15. Recall that when discussing interactive systems, it is useful to be able to identify the **user**, the **system**, **user actions** and **system feedback**. Which of the following is an example of **system feedback**?

A. a "ding" sound is made when the user clicks on a button

B. the user presses the spacebar key and nothing seems to happen

C. the system completes a task without displaying progress to the screen

D. the user scratches the back of their head

E. the system starts and nothing is displayed to the screen

F. None of the above

(1) 16. In an interactive Processing program, the `keyPressed()` function is called:

A. Only when it is invoked by the programmer via a function call

B. Continually at fixed, regular intervals

C. Whenever the user presses *any key* on the keyboard

D. Exactly once when the canvas is drawn for the first time

E. Whenever the user presses *a specific key* on the keyboard

(1) 17. What happens when the following Processing program is run?

```
def setup():
    background(215)

def draw():
    background(215)
    ellipse(mouseX, mouseY, 30, 30)
```

A. a single circle follows the mouse

B. a single circle is drawn that stays fixed in place

C. a stream of circles follows the mouse

D. nothing is drawn (but the program doesn't crash)

E. the program crashes due to an error

(1) 18. How does the following Processing program behave when it is run?

```
def setup():
    background(215)
    fill(255)

def draw():
    ellipse(mouseX, mouseY, 10, 10)
    fill(0)
    ellipse(mouseX, mouseY, 20, 20)
    background(215)
```

A. a white circle inside a larger black circle follows the mouse

B. a white circle follows the mouse

C. a black circle follows the mouse

D. nothing is drawn (but the program doesn't crash)

E. the program crashes due to an error

(1) 19. In Python, what is a *literal*?

A. a fixed value B. a compound action C. an atomic data type D. a kind of variable

(1) 20. The value `False` is an example of:

A. string data **B. Boolean data** C. float data D. integer data E. None of the listed options

(1) 21. Which one of the following is an example of *string* data?

A. 42.50 B. 92 **C. '42.50'** D. False E. 13e6 F. None of the listed options

(1) 22. What value is associated with the variable `banana` by the time the following program ends?

```
apple = 10
apple = 12
banana = apple * 1
apple = banana + banana
orange = banana + 1
apple = 0
banana = orange - apple
```

A. -11 B. 0 C. 10 D. 12 **E. 13** F. 24 G. None of the listed options

(1) 23. What is printed to the console by the following program?

```
def apply_discount(cost):
    cost = 26.5 - 10.0
    print(cost)

cost = 26.5
apply_discount(cost)
print(cost)
```

A. cost cost B. 16.5 16.5 **C. 16.5 26.5** D. 26.5 25.6 E. The program crashes

For questions 24 through 27, assume that the following variable initializations are given.

```
f1 = 1.5
f2 = 3.0
f3 = 0.0
i1 = 8
i2 = 3
i3 = 0
```

(1) 24. What is the value of the expression `8 + 3 - 2.0 * 3`?

A. 3 B. 3.0 C. 5 **D. 5.0** E. 27 F. 27.0

(1) 25. What is the value of the expression `i1 % i2 + i3`?

A. 0 B. 1 **C. 2** D. 3 E. 2.666

(1) 26. What is the value of the expression: `(0 == i1) or (i1 < 10)`

A. True B. False

(1) 27. What is the value of the expression:
`not ((i1 <= i1) and (i3 < f2) and (i1 == 8)) or (i2 > i2)`

A. True **B. False**

(1) 28. What is printed to the console by the following program?

```
def check(a):
    return max(50, min(a, 100) )

print( check(175) )
```

A. 50 **B. 100** C. 175 D. `check(175)` E. The special Python value `None`

(1) 29. What is printed to the console by the following program?

```
def message():
    m = "me want cookie!"
    return m

message()
print(m)
```

A. m

B. me want cookie!

C. me want cookie!
me want cookie!

D. The special Python value `None`

E. The program crashes due to an error

(1) 30. What is printed to the console by the following program?

```
def cookies():
    c = "COOKIES!"
    return c

c = "me hungry!"
cookies()
print(c)
```

A. c

B. me hungry!

C. COOKIES!

D. The special Python value `None`

E. The program crashes due to an error

(1) 31. In the context of Python, what is a **library**?

A. A series of webpages on the internet that describe the Python language

B. The documentation (comments and docstrings) for a specific program's code

C. An online database from which Python programs can be downloaded

D. A collection of functions that can be called from your program

E. None of the above

(1) 32. What is printed to the console by the following program?

```
ernie_votes = 7
bert_votes = 5

if bert_votes < ernie_votes :
    print("We like Ernie!")
else:
    print("We like Bert!")
```

Which best describes what this code does?

A. Only "We like Ernie!" is displayed on the console.

B. Only "We like Bert!" is displayed on the console.

C. Both "We like Ernie!" and "We like Bert!" are displayed on the console.

D. Nothing is displayed (but the program doesn't crash)

E. The program crashes due to an error

(1) 33. What is printed to the console by the following program?

```
cakes = 9
pies = 9

if (cakes == pies) :
    print("All cakes and pies are created equal.")

print("Okay, who started eating the timbits?")
```

Which best describes what this code does?

A. Only "All cakes and pies are created equal." is displayed to the console.

B. Only "Okay, who started eating the timbits?" is displayed to the console.

C. Both "All cakes and pies are created equal." and "Okay, who started eating the timbits?" are displayed to the console.

D. Nothing is displayed (but the program doesn't crash)

E. The program crashes due to an error

Section 2 - Written Answer

How to answer questions on the rest of the exam

The following questions assess your Processing programming ability. The answers should be relatively short, so don't write a lot of code for them. Some of these are "easy", some are "moderate" and some may be a little more difficult.

Each question will ask you to write code in one of three different styles:

- **a code snippet:** Just a few lines of code by themselves without any function headers.
- **a function:** A function header accompanied by a function body. You do **NOT** need to write any code that tests or otherwise uses the function by calling it.
- **a complete interactive program**: A complete program using `setup()`, `draw()`, and whatever else you need such that if you copy/pasted it into Processing, the whole program would work.

For all three styles, you do **NOT** need to write comments or docstrings unless the question explicitly asks you to do so.

We will deduct a small portion of marks for syntax errors, though we will not penalize you repeatedly for making the same mistake.

(3) 34. Write a **code snippet** that draws a blue circle inside a larger dark gray square.

```
fill(100)
rect(10, 10, 40, 40)
fill(0,0,255)
ellipse(30, 30, 20, 20)
```

Marking:

(1) calls to fill(); (1) calls to rect() and ellipse(); (1) shapes drawn in the correct order

(3) 35. Write a **complete interactive program** which initially has a white background. When the user clicks the mouse, a green circle is drawn at the current mouse location. When the user presses any key, the canvas is cleared.

```
def setup():
    background(255)

def draw():
    return

def keyPressed():
    background(255)

def mouseClicked():
    fill(0,255,0)
    ellipse(mouseX, mouseY, 30, 30)
```

Marking:

- (1) interactive function headers are correctly defined (-0.5 if both setup() and draw() not defined)
- (1) correct behaviour on mouse click
- (1) correct behaviour on key press

(3) 36. Write a **function** called `black_hole` which draws a black circle in the center of a square canvas. The size of the canvas and the diameter of the circle should both be separate parameters to the function. For this question, include a **docstring** with your function.

```
def black_hole(canvas, d):
"""
draws a black circle centered in the middle of the canvas, which is assumed to be square
canvas: integer, indicates the size of the canvas
d: integer, the diamter of the black circle
"""
    fill(0)
    ellipse(canvas/2, canvas/2, d, d)
```

Marking:

- (1) function header correct, including parameters
- (1) function body correct
- (1) docstring is appropriate and adequately describes parameters

(3) 37. Write a **function** that has one integer **parameter**, and **returns** a Boolean value that is `True` if the value of the parameter is **positive** and `False` otherwise. Recall that 0 is NOT positive.

```
def positive(i):
    if i > 0:
        return True
    else:
        return False
```

Marking:

- (1) function header
- (1) calculation (may use properly initialized local variable)
- (1) return

(4) 38. Answering this question will require two steps:

Step 1: Write a **function** called `word_lengths` which accepts three strings as parameters. The function should compute and return the sum of the lengths of all three strings. For example, if the first string is "hi", the second string is "yes", and the third string is "no", the combined length of the strings is 7. **Hint:** The Python function `len(s)` returns the length of its argument `s`.

Step 2: Write a **code snippet** that calls your `word_lengths` function three times, each time using a different set of string values of your choice as arguments. Your code should also print the returned value of each of the three function calls to the console.

```
def word_lengths(w1, w2, w3):
    return len(w1 + w2 + w3)

print(word_lengths("hi", "hello", "goodbye"))
print(word_lengths("milk", "cookies", "chocolate"))
print(word_lengths("ernie", "bert", "elmo"))
```

Marking:

- (1) word_lengths computes right answer
- (1) function uses return correctly
- (1) function calls are correct
- (1) values are correctly printed

(5) 39. Write a **complete interactive program** in which **two circles follow the mouse** on a 300x300 canvas. The two circles should be of different sizes, with the smaller one inside the bigger one.

When the user presses the + key, the size of the **outer circle** should increase (the size of the inner circle **should NOT change**). When the user presses the − key, the size of the **outer circle** should decrease. Your code must make sure that the outer circle never becomes smaller than the inner circle.

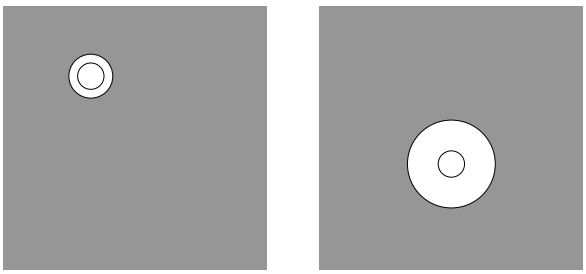

Figure 1: Two circles following the mouse, with the outer circle changing size from key presses

```
inner_size = 30
outer_size = 50

def setup():
    size(300, 300)
    background(150)

def draw():
    global inner_size, outer_size
    background(150)
    ellipse(mouseX, mouseY, outer_size, outer_size)
    ellipse(mouseX, mouseY, inner_size, inner_size)

def keyPressed():
    global outer_size, inner_size

    if key == "+":
        outer_size += 5
    if key == "-":
        outer_size = max(outer_size - 5, inner_size)
```

Marking: (1) background drawn correctly (1) variable used to represent outer circle size (1) two circles with different sizes correctly follow the mouse (1) well-formed if statements in keyPressed (1) if-statements have correct behaviour