THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

This exam paper must not be removed from the venue

Venue _____

Seat Number _____

Student Number |__|__|__|__|__|__|__|__|

Family Name _____

First Name _____

School of Information Technology and Electrical Engineering

EXAMINATION

Semester Two Final Examinations, 2020

CSSE1001/7030 Introduction to Software Engineering

This paper is for St Lucia Campus students.

Examination Duration: 120 minutes

Reading Time: 10 minutes

Exam Conditions:

This is a Central Examination

This is a Closed Book Examination - no materials permitted

During reading time - write only on the rough paper provided

This examination paper will be released to the Library

Materials Permitted In The Exam Venue:

(No electronic aids are permitted e.g. laptops, phones)

A FX82 series calculator or a university approved and labelled calculator or a calculator from the university's list of approved calculators

Materials To Be Supplied To Students:

1 x Multiple Choice Answer Sheet

Instructions To Students:

Additional exam materials (eg. answer booklets, rough paper) will be provided upon request.

For Examiner Use Only

Question	Mark

Total _____

For all questions, please choose the most appropriate answer if it appears that more than one option is a potentially correct answer. All coding questions relate to the Python 3 programming language. If an evaluation produces an error of any kind, choose Error as your answer. Different questions may have different numbers of choices. Each question is worth one mark.

1. What does the expression `5.1 + 12.2 // 2` evaluate to?

   a) `11`
   b) `11.0`
   c) `11.1`
   d) `Error`

2. What does the expression `5.0 - 3 % 2` evaluate to?

   a) `4`
   b) `4.0`
   c) `0`
   d) `-4`
   e) `Error`

3. What does the expression `5.0 / 2 % 2` evaluate to?

   a) `0.5`
   b) `2.0`
   c) `2`
   d) `Error`
   e) `None of the other choices are correct`

4. What does the expression `(2,9) + ([7])` evaluate to?

   a) `(2,9,[7])`
   b) `(2,9,([7]))`
   c) `(2,9,7)`
   d) `Error`
   e) `None of the other choices is correct`

5. What does the expression `['grue'] <= ['cat']` evaluate to?

   a) `True`
   b) `False`
   c) `['grue']`
   d) `Error`
   e) `None of the other choices is correct`

6. What is the value of `a` after the following statements are evaluated?

```
x = [3, '7', '\n d r'.split()]
a = x[2][2]
```

a) `'d'`

b) `'r'`

c) `'d r'`

d) `Error`

e) `None of the other choices is correct`

7. What is the result of `max(1.2,4//3) < min(7-2**3,1.1)`?

a) `True`

b) `False`

c) `Error`

d) `None of the other choices is correct`

8. What is the value of `a` after the following statements are evaluated?

```
x = ['a', (3,4), {1:'b'}]
y = x
y[2] = {2:'c'}
a = x[2][1]
```

a) `'b'`

b) `'c'`

c) `Error`

d) `None of the other choices are correct`

9. After the assignment `s1 = "Never give up"`, which of the following statements assigns `'r g'` to `s2`?

a) `s2 = s1[4:7]`

b) `s2 = s1[4:8]`

c) `s2 = s1[-9:-7]`

d) `s2 = s1[-9:-6]`

e) `More than one of the other choices are correct`

10. After the assignment `s1 = "Never give up"`, what will the value of `y` be after the following command is entered?

```
s2=s1[6::-2]
```

a) `s2 = 'p v'`

b) `s2 = 'give '`

c) `s2 = 'grvN'`

d) None of the other choices are correct

11. What will be in printed out after the following code is executed?

```
for i,j in (['fr','qt'],):
    j,i=i,j
    print(i,j)
```

a) `fr qt`

b) `qt fr`

c) `f r`
   `q t`

d) `r f`
   `t q`

e) None of the other choices is correct

12. Consider the following code:

```
x = input("Please enter a number: ")
x1 =  x[:-1]
print("The number you entered was:", x1)
```

What will be printed after the above code is executed, assuming that you enter 236 when prompted?

a) The number you entered was: 236

b) The number you entered was: 23

c) The number you entered was: 632

d) An error message

13. What is the value of `d2` after the following statements are evaluated?

```
d1 = {1:'c', 2:'d', 3:'e'}
d2=d1.update({4:['f']})
```

d

a) `{1:'c', 2:'d', 3:'e'}`

b) `{1:'c', 2:'d', 3:'e', 4:['f']}`

c) `{1:'c', 2:'d', 3:'e', 4:'f'}`

d) None

e) None of the other choices is correct

14. After executing the code below, what would be the contents of `a`?

```
a={1:"s",2:"t",3:"r"}
b={4:"i",5:"n"}
a.update({6:b.get(7)})
```

b

a) `{1:'s', 2:'t', 3:'r'}`

b) `{1: 's', 2: 't', 3: 'r', 6: None}`

c) `{1: 's', 2: 't', 3: 'r', 6: []}`

d) Error

e) None of the other options is correct

15. What will be printed out when the following code is run?

```
def g(w):
    w.append(f)
    return w
f=9
w=[60]
z = g(w)
print(z,w)
```

[60, 9] [60, 9]

a

a) `[60, 9] [60, 9]`

b) `[60, 9] [60]`

c) `[60, 9] 9`

d) An error message will be printed out because f is not passed to the function as an argument

16. The following recursive function definition is used in this question and the next one.

```
def w(y) :
    if y == 1 :
        return y
    y -=1
    return w(y-1) - 1
```

What will be returned when w(4) is called?

a) 0

b) 1

c) 5

d) An error message indicating a RecursionError due to maximum recursion depth being exceeded

17. What will be returned when the function w(1) is called?

a) 0

b) 1

c) -1

d) An error message indicating a RecursionError due to maximum recursion depth being exceeded

18. The following class and method definitions are used for this and the following 4 questions.

```
class A(object) :
    def __init__(self, x) :
        self._x = 2 * x
    def m1(self, x) :
        return self.m2(x) + 2
    def m2(self, x) :
        return x - 1

class B(A) :
    def m2(self, y) :
        self._y = y
        return self._x + self._y

class C(B) :
    def __init__(self, x, y) :
        super().__init__(x)
        self._y = y + 2
    def m1(self, x) :
        return self._x + self._y

class D(B) :
    def __init__(self, x, y) :
        super().__init__(x)
        self._x += y
        self._y = y + 2
    def m1(self, y)  :
        return self._y + y
    def m2(self, x) :
        return super().m2(x) - x

a = A(1)
b = B(2)
c = C(1, 1)
d = D(2, 1)
```

What does `a.m1(2)` return?

a) 1

b) 2

c) 3

d) 4

e) None of the other choices is correct

19. What does `b.m2(1)` return?

a) 5
b) 6
c) 7
d) 8
e) None of the other choices is correct

20. What does `c.m2(3)` return?

a) 3
b) 4
c) 5
d) 6
e) None of the other choices is correct

21. What does `d.m1(3)` return?

a) 5
b) 6
c) 7
d) 8
e) None of the other choices is correct

22. What does `d.m2(1)` return?

a) 5
b) 6
c) 7
d) 8
e) None of the other choices is correct

23. What is the value of `y` after the following has been evaluated?

```
z = lambda v,w: v*w
xs = [1,2,3,4]
ys = [3,4,5,6]
y = [z(v,w) for v in xs for w in ys if w<4]
```

a) [3,6,9,12]

b) [1,2,3,4]

c) [3,4,5,6,6,8,10,12,9,12,15,18]

d) None of the other choices is correct

24. After running the following code:

```
import random
xs=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
red_list=[(x,random.random()) for x in xs[::2]]
red_list.sort()
z=[ (y,x) for y,x in red_list]
```

which of the following represents the most plausible contents of z?

a) [(0.1533077540075557, 'c'), (0.5043464158983711, 'd'), (0.03274734535677237, 'e'), (0.37223199582110045, 'f'), (0.8592305511956286, 'g'), (0.6097854587147008, 'h')]

b) [('c', 0.4038438875471476), ('d', 0.8485458668524117), ('e', 0.3460401744304016), ('f', 0.9804209881137512), ('g', 0.11978067181916108), ('h', 0.5556118897608778)]

c) [(0.23302650595515995, 'h'), (0.4433426407371851, 'e'), (0.49198326769565137, 'd'), (0.621306659920582, 'f'), (0.7386322758139315, 'c'), (0.7559255166995879, 'g')]

d) [('a', 0.002095459487238327), ('c', 0.2040937365702672), ('e', 0.6762360347828238), ('g', 0.9745492957346064)]

e) Error

25. The following partial definition of a SwimRecord class is used in this and the following two questions.

```
class SwimRecord(object) :
    def __init__(self, name, club) :
        """Parameters:
        name(str): swimmer's name
        club(str): swimmer's club
        self._swim_record(dict): Data record to store
        swim meets and swimtimes. The key is the name of
        the swim meet; the value is the time recorded for
        each swim meet"""
        self._name = name
        self._club = club
        self._swim_record = {}

    def update_swim_record(self, new_results: dict) :
        """Add results from 'new_results' into record."""
        self._swim_record.update(new_results)

    def drop_item(self, swim_meet):
        """ Remove from the record the item whose key is
        swim_meet"""
        ## code block 1 ##

    def get_swim_results(self, swim_meet: str) :
        """Get swim results."""
        return self._swim_record.get(swim_meet,'Err')

    def get_swim_times(self) :
        """Return all swim meets in a list"""
        ## code block 2 ##
```

What is the required code for **## code block 1 ##**?

a) `self._swim_record -= self._swim_record[swim_meet]`

b) `self._swim_record.pop(swim_meet)`

c) `self._swim_record[swim_meet]=None`

d) None of the other choices is correct

26. What is the required code for **## code block 2 ##**?

a) `return [i for i,j in self._swim_record.items()]`

b) `return self._swim_record`

c) `return list(self._swim_record.values())`

d) `return swim_record.update(self)`

e) More than one of the other choices are correct

27. Assume that an object from the SwimRecord class has been created and has the name `Mary_Brown`. Assume also that it contains an accurate record of the past year's swim results, including those for the `'Southport'` swim meet. Which of the following will return the swim results for the `'Southport'` swim meet for `Mary_Brown`?

    a) `Mary_Brown.get_swim_results('Southport')`

    b) `Mary_Brown._swim_record('Southport')`

    c) `Mary_Brown._swim_record[get_swim_results('Southport')]`

    d) `More than one of the above`

    e) `None of the above.`

28. What does the following evaluate to, assuming `d={1:'Killer rabbit'}`?

```
d[1].join('huge teeth'.split())
```

    a) `'huge teeth Killer rabbit'`

    b) `'hugeKiller rabbitteeth'`

    c) `'huge Killer rabbit teeth'`

    d) `Error`

    e) `None of the other choices is correct`

29. This and the following two questions refer to the following function definition, which is missing three lines of code. The function reads raw wind speed data from a file and calculates the average and maximum wind speeds for each day. The following is an example of a data file (wind_speed.csv).

```
11,12,7,20,2,15.8
21,13.2,14.4,25,10,12.4
11,15,6.2
```

Each line of the file contains the wind speed readings collected on a single day. These values may be integer or floating point and are separated by a comma. Each day may have different numbers of readings. The average wind speed for a day is simply the average of all the values on the line. The maximum wind speed for a day is the largest value on the line. The results are written to an output file in the same order in which they are read from the input file. The logic assumes that the data in the input file is in the correct format. The definition of the process function, with three missing lines, is given below.

```
def process(wind_speeds, processed_data) :
    with open(wind_speeds, "r") as wind_data, \
          open(processed_data, "w") as result_data :
       for day in wind_data:
           day = day.strip()
           wind_speeds = day.split(',')
           ##line 1: Set initial max wind speed##
           total = 0
           ##line 2: Calculate average & max wind speeds##
           ##line 3: Update max wind speed##
           average_wind_speed = total / len(wind_speeds)
           result_data.write(f"{str(average_wind_speed)},"
                             + f"{str(max_wind_speed)}\n")
```

The result of calling the completed function on the file described above, for example by:
process('wind_speed.csv', 'processed_results.csv')
would result in the following data being saved to processed_results.csv.

```
11.3,20.0
16.0,25.0
8.5,15.0
```

When answering the following two questions assume that the correct code has been implemented from the previous question(s).

What code is required at ##line 1: Set initial max wind speed.##?

a) `max_wind_speed = wind_speeds`

b) `max_wind_speed = wind_speeds[0]`

c) `max_wind_speed = float(wind_speeds)`

d) `max_wind_speed = float(wind_speeds[0])`

e) None of the other options is correct

30. What is the required code for ##line 2 : Calculate average and max wind speeds##?

a)
```
for wind_speed in day :
    wind_speed = float(wind_speed)
    total += wind_speed
```

b)
```
for wind_speed in day :
    total += wind_speed
```

c)
```
for wind_speed in wind_speeds :
    wind_speed = float(wind_speed)
    total += wind_speed
```

d)
```
for wind_speed in wind_speeds :
    total += wind_speed
```

e) None of the other options is correct

31. What is the required code for ##line 3: Update max wind speed##?

a)
```
if wind_speed > max_wind_speed :
    max_wind_speed = wind_speed
```

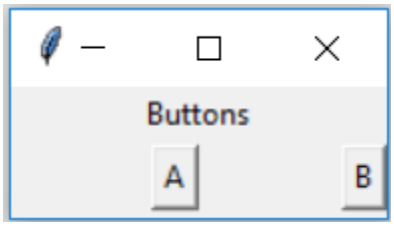
b)
```
if wind_speeds[0] > max_wind_speed :
    max_wind_speed = wind_speeds[0]
```

c)
```
if float(wind_speed) > float(max_wind_speed) :
    float(max_wind_speed) = float(wind_speed)
```

d) `max_wind_speed = max(wind_speed)`

e) More than one of the other options is correct

32. This and the following question relate to the following partial definitions. In a GUI application we decide we need a widget that contains two buttons and that this widget is to appear within the main window of the application below the label as shown in the image below.




```
class ButtonsFrame(tk.Frame) :
     def __init__(self, parent) :
         tk.Frame.__init__(self, parent.root)
         b1 = tk.Button(self, text= "A")
         b2 = tk.Button(self, text = "B")
          ## lines 1 and 2 ##

class MainWindow(object) :
      def __init__(self, root) :
          self.root = root
          tk.Label(root, text="Buttons").pack()
          bf = ButtonsFrame(self)
          ## line 3 ##
```

What is the required code for `## lines 1 and 2 ##`?

a)
```
b1.pack(side=tk.LEFT, expand=1)
b2.pack(side=tk.LEFT)
```

b)
```
b1.pack(side=tk.LEFT, expand=1)
b2.pack(side=tk.LEFT, expand=1)
```

c)
```
b1.pack(side=tk.LEFT, fill=tk.BOTH)
b2.pack(side=tk.LEFT, fill=tk.BOTH)
```

d)
```
b1.pack(side=tk.LEFT, fill=tk.BOTH)
b2.pack(side=tk.LEFT, fill=X)
```

e) More than one of the above is correct

33. What is the required code for `## line 3 ##`?

a) `bf.pack(expand=1)`

b) `bf.pack(fill=tk.BOTH, expand=1)`

c) `bf.pack()`

d) `bf.pack(fill=tk.BOTH)`

e) More than one of the above is correct

34. What will happen when the following code is executed?

```
import tkinter as tk

root=tk.Tk()
root.after(50,root.destroy)
root.mainloop()
```

a) A window will be opened and then close after 50 seconds

b) A window will be opened and closed after 0.05 seconds

c) A window will be opened and then closed after 5 seconds. After this a different window called mainloop will be opened

d) An error message will be generated

35. Which of the following is true?

a) Strings, integers, floats, booleans and tuples are all immutable

b) Objects from user defined classes are by default immutable

c) Dictionaries are immutable

d) More than one of the above options are true

e) None of the other options are true

36. What will be returned after the following commands are entered?

```
import operator
print(sum(tuple(map(operator.add, [3,4,5,6],[4,5,6,7]))))
```

a) 40

b) (18,22)

c) (7,9,11,13)

d) <map object at 0x0370C3D0>

e) Error

37. Assume `y = [3,4,5]` and `z= [1,2]`. What will be popped from y after the following statement is executed?

```
y.pop(z.extend([1,2])[2])
```

a) 4

b) 5

c) 3

d) Nothing. An error message will be produced

38. What will be the value of `x` after evaluating these statements?

```
x = [4, 5, 6, 7]
x.extend(x)
x.insert(1, x.pop(3))
```

a) `[4,7,5,6,[4,5,6,7]]`
b) `[4, 7, 5, 6, 4, 5, 6, 7]`
c) `None`
d) `Error`
e) `None of the other options is correct`

39. This and the following question refer to the following definition:

```
def g(x, z):
    x.append(z)
    return x
```

What is the value of y after the following is evaluated?

```
y = ['a', 'b', 'c']
g(y, 'd').extend(g(y[:], 'd'))
```

a) `['a', 'b', 'c', 'd']`
b) `['a', 'b', 'c', 'd', 'd']`
c) `['a', 'b', 'c', 'd', 'a', 'b', 'c', 'd']`
d) `['a', 'b', 'c', 'd', 'a', 'b', 'c', 'd', 'd']`

40. What is the value of y after the following is evaluated?

```
y = ['a', 'b', 'c']
g(y[:], 'd').extend(g(y, 'd'))
```

a) `['a', 'b', 'c', 'd']`
b) `['a', 'b', 'c', 'd', 'd']`
c) `['a', 'b', 'c', 'd', 'a', 'b', 'c', 'd']`
d) `['a', 'b', 'c', 'd', 'a', 'b', 'c', 'd', 'd']`

END OF EXAMINATION