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

Semester One Examinations, 2022

CSSE1001/7030 Introduction to Software Engineering

This paper is for St Lucia Campus students.

Examination Duration: 120 minutes

Planning Time: 10 minutes

Exam Conditions:

•This is a Closed Book examination - no written materials permitted
•Casio FX82 series or UQ approved or labelled calculator only
•During Planning Time - Students are encouraged to review and plan responses to the exam questions
•This examination paper will be released to the Library

Materials Permitted in the Exam Venue:
(No electronic aids are permitted e.g. laptops, phones)

None

Materials to be supplied to Students:
Additional exam materials (e.g. answer booklets, rough paper) will be provided upon request.

1 x Multiple Choice Answer Sheet

Instructions to Students:
If you believe there is missing or incorrect information impacting your ability to answer any question, please state this when writing your answer.

Answer all questions on the MCQ sheet provided. All questions are equally weighted.

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 `12 + 4.2 // 2` evaluate to?

   a) `14.1`
   b) `12.2`
   c) `14`
   d) `14.0`
   e) `None of the above`

2. What does the expression `-1 <= -1 >= -2` evaluate to?

   a) `0`
   b) `1`
   c) `True`
   d) `False`
   e) `Error`

3. What does the expression `(11 - 6) % -3` evaluate to?

   a) `-1`
   b) `1`
   c) `-2`
   d) `2`
   e) `None of the above`

4. What does the expression `bool(not('c') and 'b')` evaluate to?

   a) `'b'`
   b) `True`
   c) `False`
   d) `Error`

5. What does the expression `'hard to know'.split()[-1]` evaluate to?

   a) `['hard', 'to']`
   b) `'know'`
   c) `'hard'`
   d) `['to', 'know']`
   e) `None of the above`

6. What will be printed out when the following code is executed?

```
for i,c in enumerate(['to','do']):
    print(c,i)
```

a) `ot`
   `od`

b) `to 0 do 1`

c) `to 0`
   `do 1`

d) `ot 0`
   `od 1`

e) None of the other choices are correct

7. The following function prompts a user to repeatedly enter numbers until a 0 is entered, and then does some processing. Which of the following descriptions best describes the purpose of the function below?

```
def routine() :
    t = 0
    r = int(input('Please input an integer: '))
    while r != 0 :
        if r // 5 == 0:
            t += r
        r = int(input('Please input an integer: '))
    return t
```

a) It returns the sum of all integers entered which are less than 5.

b) It returns the sum of the first 5 integers entered.

c) It returns the sum of all integers entered, provided they are not divisible by 5.

d) It is an infinite loop which is never exited.

e) It returns the sum of all integers entered that are divisible by 5.

8. After the assignment `s1 = "Strategic initiative"` which of the following statements assigns `"ate"` to `s2`?

a) `s2 = s1[3:6]`

b) `s2 = s1[-17:6]`

c) `s2 = s1[3:-14]`

d) `All of the above`

e) None of the above

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

```
def g():
    w.append(9)

w = [4, 6]
g()
print(w)
```

a) [4, 6, 9]

b) [4, 6]

c) None

d) Error

e) None of the above

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

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

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 are correct

11. What is the value of a after the following code is executed?

```
a = 1
b = 2
while a < 3:
    a *= b
    b += 1
```

a) 4

b) 6

c) 10

d) The code produces an error

e) None of the above

12. Some code for a simple calculator is needed to enable a user to add, subtract, multiply or divide two numbers. The user must be able to do calculations as often as they wish. A snippet of the functioning calculator is given below:

```
Select operation.
1.Add; 2.Subtract; 3.Multiply; 4.Divide
Enter operation(1/2/3/4): 3
Enter first number: 4
Enter second number: 3
4.0 * 3.0 = 12.0
Enter operation(1/2/3/4): 2
```

The partially completed code to implement the calculator is provided below:

```
print("""Select operation.
1.Add; 2.Subtract; 3.Multiply; 4.Divide""")

## Line 1 ##
    # prompt type of operation from user
    select = int(input("Enter operation(1/2/3/4): "))
    # prompt for input numbers and perform calculation
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    ### Line 2 ###
    print(num1, x[select][0],num2, "=", x[select][1])
```

What code is required for **`## Line 1 ##`**?

a) `for x in '1234':`

b) `for x in range(1,5):`

c) `while x in [1,2,3,4]:`

d) `while True:`

e) `More than one of the above options are suitable`

13. Which of the following would be suitable for **`## Line 2 ##`** of the code in Question 11?

a) `x=[(1,'+',num1+num2),(2,'-',num1-num2),`
`(3,'*',num1*num2),(4,'/',num1/num2)]`

b) `x=((1,'+',num1+num2),(2,'-',num1-num2),`
`(3,'*',num1*num2),(4,'/',num1/num2))`

c) `x={1:('+',num1+num2),2:('-',num1-num2),`
`3:('*', num1*num2),4:('/',num1/num2)}`

d) `x={1:['+',num1+num2],2:['-',num1-num2],`
`3:['*',num1*num2],4:['/',num1/num2]}`

e) More than one of the above choices would be suitable

14. What is printed after calling the function `f1`?

```
def f1() -> None:
    v = 6
    x = f2(v)
    print(v)

def f2(x: int) -> int:
    v = 3
    return x // v

f1()
```

a) 6

b) 4

c) 3

d) 0

e) Error

15. What will be printed after the following code is executed, assuming `23` is the two digit number entered:

```
x = input("Please enter a two digit number: ")
x1 = int(x)
x1 = x1[0]
print("The first digit is:", x1)
```

a) The first digit is: 2
b) The first digit is: 23
c) The first digit is:
d) An Error message
e) None of the above

16. What is printed after the following code is executed?

```
def f(x: list[int]) -> float:
    y = 0
    z = 0
    for i in x:
        y += i
        z += 1
    return y / z

a = [10, 20, 30, 40]
print(f(a))
```

a) 10.0
b) 25.0
c) 33.33333333336
d) 40.0
e) Error

17. This and the following two questions refer to the function, load_anag, which is missing three lines of code. The function reads in words from a file, with each line of the file containing one word and possibly some additional whitespace. The function then creates a dictionary of anagrams (i.e. a group of words with the same letters) from the file.

    For example, if the file contained the words `'edit','tide','diet','trams','smart'`, then the resulting dictionary would be:

    `{'deit':['edit', 'tide', 'diet'], 'amrst': ['trams','smart']}.`

    Note that each value in the dictionary is a list of unique anagrams. Each key is a string containing the same letters as the words in the value, but with the letters in alphabetical order.

```
def load_anag(filename):
    """ Add anagrams from filename to a dictionary, d, and
    return d.
     load_anag(str) => dict(str: list(str))
    """
    d={}
    fd=open(filename, 'r')
    for line in fd:
        ## line 1 ## remove whitespace
        key=sort_str(word)#sort_str()alphabetically sorts a string
        value=d.get(key)
        ## line 2 ##
           d[key]=[word]
        ## line 3 ##
            value.append(word)
    return d
    fd.close()
```

What code is required for ## line 1 ##?

a) `word=line.split()`

b) `word=line.strip()`

c) `word=line.split(' ')`

d) More than one of the above options are suitable

18. Which of the following would be suitable for ## line 2 ##?

a) `if word not in value:`
b) `if d[key]=value:`
c) `if value is None:`
d) `None of the other choices are suitable`

19. Which of the following would be suitable for ## line 3 ##?

a) `elif word not in value:`
b) `elif word != '':`
c) `elif len(value)>1:`
d) `None of the other options are suitable`

20. The following recursive function definition is used in the next two questions.

```
def fn1(x):
    if len(x) == x[0] :
        return x
    return fn1(x[3:] + [x[0]])
```

What will the function call `fn1([2, 4, 1, 1, 4])` return?

a) `[5, 2, 1, 4, 1]`
b) `[2,5]`
c) `[1]`
d) `RecursionError will be raised due to maximum recursion depth being exceeded.`

21. What will the function call `fn1([3, 1, 3, 2, 7])` return?

a) `[7, 4, 1, 3, 2]`
b) `[7, 1, 2]`
c) `[3, 7]`
d) `RecursionError will be raised due to maximum recursion depth being exceeded.`

22. What is the value of zs after running the following code:

```
f = lambda x, y : x - y
g = lambda x : x*2
zs = [g(x) for x in [1,2,3] if f(x,2)]
```

a) `[2, 6]`
b) `[1, 9]`
c) `[2]`
d) `[]`
e) `None of the above`

23. What is the purpose of the raise statement in Python?

a) To attempt to execute a block of code and handle at least some of the errors that may be caused by the statements in the block of code.

b) To provide an error handling function that will be called if any error occurs in a block of code.

c) To indicate that the code has encountered an error it cannot handle locally.

d) To identify potential errors that may be encountered in a block of code.

24. We want to store the "maximum team size" as global constant in a Python program. According to standards, which name is most appropriate for this variable?

```
a) maximum_team_size
b) MaximumTeamSize
c) __Maximum_Team_Size__
d) MAXIMUM_TEAM_SIZE
e) None of the above.
```

25. The following is a recursive function to calculate the sum of a list of numbers. Example usage:

```
sum([]) -> 0
sum([1])-> 1
sum([1, 2]) ->  3 # 1 + 2
sum([1, 2, 3]) -> 6 # 1 + 2 + 3
def sum(nums) :
    if len(nums) == 0 :
        return 0
    elif len(nums) == 1:
        return nums[0]
    return ## TODO: what goes here
```

Which code fragment will correctly complete the function above?

```
a) (sum(nums[:len(nums) // 2]) +
    sum(nums[len(nums) // 2:]))

b) (sum(nums[1:len(nums) / 2]) +
    sum(nums[len(nums) / 2:-1]))

c) sum(nums[1:len(nums)]) + sum(nums[len(nums):-1]

d) sum(nums[1:]) + sum(nums[:-1])
```

26. What is the value of x after the following is evaluated

```
d = {'Brown' : {'ID': 732, 'Orders': ['chisel', 'spanner']},
'Black': {'ID': 461, 'Orders':
['lathe', 'crowbar']}}
x = d.get('White',{}).get('Orders',[])
```

a) `['chisel', 'spanner']`
b) `[]`
c) `['lathe', 'crowbar']`
d) `Error`
e) `None of the above`

27. What is the value of x after the following is evaluated?

```
d = {'Brown' : {'ID': 732, 'Orders': ['chisel', 'spanner']},
'Black': {'ID': 461, 'Orders':
['lathe', 'crowbar']}}
x = d.get('Brown',{}).get('Orders',[]).append('hammer')
```

a) `['chisel', 'spanner', ['hammer']]`
b) `['chisel', 'spanner', 'hammer']`
c) `None`
d) `Error`
e) `None of the above`

The following partial definition of a SwimRecord class is used in the following three questions.

```
class SwimRecord():
    def __init__(self, name, club):
        """Parameters:
        name(str): swimmer's name
        club(str): swimmer's club
        swim_record(dict): record of swim times"""
        self._name = name
        self._club = club
        self._swim_record = {}

    def update_swim_record(self, new_results: dict) :
        """Add the dictionary, 'new_results', into the current
        dictionary of swim results."""
        ## code block 1 ##

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

    def get_personal_best(self) :
        """Get the shortest time from all swim meets in the
        swim_record. Assume all swim meet events are over the same
        distance."""
        ## code block 2 ##
```

28. What is the required code for `## code block 1 ##`?

a) `self._swim_record += new_results`
b) `self._swim_record.update(new_results)`
c) `self._swim_record.append(new_results)`
d) None of the code blocks above are correct.

29. What is the required code for `## code block 2 ##?`

```
a) return min(self._swim_record.items())
b) return min(self._swim_record.values())
c) return min(self._swim_record.keys())
d) return self._swim_record.pop()
e) More than one of the above are suitable
```

Assume that an object from the `SwimRecord` class has been created and has the name `john_sullivan`. Assume also that it contains an accurate record of the past year's swim results, including those for the `'Noosa'` swim meet.

30. Which of the following will return the swim results for the `'Noosa'` swim meet for John Sullivan?

```
a) john_sullivan.get_swim_results('Noosa')
b) john_sullivan._swim_record('Noosa')
c) john_sullivan._swim_record['Noosa']
d) More than one of the above
e) None of the above.
```

The next five questions refer to the following class definitions and object instantiations.

```
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)
```

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

a) 1

b) 2

c) 3

d) 4

e) None of the other choices is correct

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

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

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

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

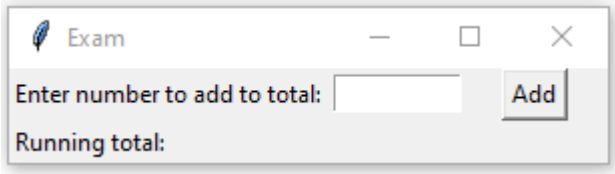
34. What does `d.m1(3)` return?

a) 4
b) 5
c) 6
d) 7
e) None of the other choices is correct

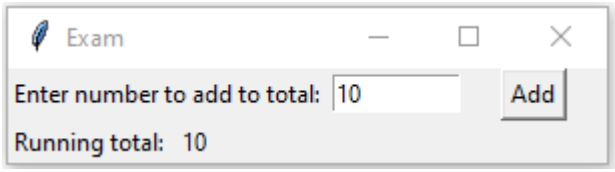
35. What does `d.m2(1)` return?

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

The next two questions relate to the following simple GUI application. The application has a text field into which users can enter numbers. When started, the GUI appears as in the image below.



When the user presses the Add button, the value in the text field is added to the running total and displayed, as shown in the image below.



The code, with two missing code fragments, is provided below.

```
import tkinter as tk

class Input(tk.Frame):
    def __init__(self, parent: tk.Tk, add_event_handler):
        super().__init__(parent)

        prompt = tk.Label(self, text="Enter number to add to
total: ")
        prompt.pack(side=tk.LEFT)

        self._entry = tk.Entry(self, width=10)
        self._entry.pack(side=tk.LEFT)

        ## Fragment 1 ##
        add_button.pack(side=tk.LEFT, padx=20)

    def get_input(self) -> int:
        return int(self._entry.get())


class Add:
    def __init__(self, master: tk.Tk):
        master.title("Exam")
        self._total = 0

        self._input_frame = Input(master, self.add)
        self._input_frame.pack(side=tk.TOP)

        total_label = tk.Label(master, text="Running total: ")
        total_label.pack(side=tk.LEFT)
        self._result = tk.Label(master)
        self._result.pack(side=tk.LEFT)

    def add(self) -> None:
        ## Fragment 2 ##
```

36. What is the required code for **`## Fragment 1 ##`**?

a) `add_button = tk.Button(self, text="Add", command=Add())`

b) `add_button = tk.Button(self, text="Add", command=Add.add)`

c)
```
add_button = tk.Button(self, text="Add",
command=self.get_input)
```

d)
```
add_button = tk.Button(self, text="Add",
command=add_event_handler)
```

e) None of the code fragments would implement the GUI correctly.

37. What is the required code for **`## Fragment 2 ##`**?

a)
```
self._total += self.get_input()
self._result.config(text=str(self._total))
```

b)
```
self._total += self._input_frame.get_input()
self._result.config(text=str(self._total))
```

c)
```
self._total += self._input_frame._entry.get()
self._result.config(text=str(self._total))
```

d)
```
self._total += self._input_frame.get_input()
self._result.config(text=str(self._input_frame.get_input()))
```

e) None of the code fragments would implement the GUI correctly.

38. The following code reverses a list and removes its negative elements.

```
def reverse_and_remove(xs: list[int]) -> list[int]:
    ans = []
    for k in range(len(xs)):
        ans.append(xs[-k-1])
    return [x for x in ans if x >= 0]
```

Let `n = length(xs)`. The complexity of `reverse_and_remove` is:

a) O(1)
b) O(n)
c) O(n^2)
d) O(log n)
e) None of the above

39. What error will this function throw when invoked?

```
def foo():
    xs = (1, 2)
    ys = [3, 4]
    return xs + ys
```

a) NameError.
b) ValueError.
c) TypeError
d) IndexError.
e) SyntaxError.

40. Consider the following function:

```
def bar(x, y, z):
    if length(x) > z[0]:
        if y in z:
            print(y)
    return y[0]
```

Which (if any) is a possible type-contract for bar?

a) `def bar(x: list[str], y: str , z: list[int]) -> None`
b) `def bar(x: int, y: list[str], z: int) -> int`
c) `def bar(x: str, y: str, z: str) -> str`
d) `def bar(x: list[str], y: str , z: list[int]) -> str`
e) None of the above.

END OF EXAMINATION