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, 2023
CSSE1001 / CSSE7030 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 and 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 Gradescope Bubble 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.

For Examiner Use Only

Question	Mark

Total _____

`Error` is the correct answer for any question with code that throws an error of any kind.

Multiple Choice

Question 1. [1 MARK]

Which of the following *cannot* be a *key* in a dictionary.

A. `'123'`

B. `(1, 2, 3)`

C. `123`

D. `[1, 2, 3]`

Question 2. [1 MARK]

Suppose the following functions have been defined.

```
1 def foo(n):
2     return n + n
3
4 def bar(n):
5     print(n + n)
```

Which of the following expressions will cause an error?

A. `z = foo(3)`

B. `z = bar(3)`

C. `z = 2 * foo(3)`

D. `z = 2 * bar(3)`

Question 3. [1 MARK]

Recall `def foo(x: int) -> int:` is type-hinted whereas `def bar():` is not. What statement is *true* about type-hints (i.e. type contracts).

A. Type-hints signal the user of the expected input to a function.

B. Type-hints are enforced. That is, if you pass a function a value with a different type than what is type-hinted Python will throw an error.

C. Python will throw an error if a function is not type-hinted.

D. None of the above.

Question 4. [1 MARK]

Suppose we want to define a name for *maximum volume* that is intended to be private. Which name is most appropriate?

A. `__maximum_volume__`

B. `MaximumVolume`

C. `_maximum_volume`

D. `MAXIMUM_VOLUME.`

Question 5. [1 MARK]

Suppose the following function definition has been made.

```
1  def foo(x):
2    if x == 1:
3        return x
4
5    return foo(x-1) * x
```

What will `foo(0)` return?

A. `-1`

B. `0`

C. `1`

D. `Error`

Question 6. [1 MARK]

Suppose the following has been executed by Python.

```
1  xs = [1, 2, 3, 4, 5, 6]
2  ys = xs[-3:-1]
```

What is stored in `ys`?

A. `[4, 5]`

B. `[4, 5, 6]`

C. `[5, 4]`

D. `[]`

Question 7. [1 MARK]

Which option will throw an `IndexError` in the following code when replacing `#sub`?

```
1  xs = [0, 1, 2, 3]
2  #sub
```

A. `xs[-len(xs)]`

B. `xs[1-len(xs)]`

C. `xs[-1-len(xs)]`

D. `xs[len(xs)-1]`

Question 8. [1 MARK]

What is the purpose of the `bind()` method in `tkinter`?

A. To create a new widget.

B. To add an event handler to a widget.

C. To remove a widget from a window.

D. To resize a widget.

Question 9. [1 MARK]

What is stored in `count` after the following is evaluated?

```
1 count = 0
2 for x in "abcdef":
3     if x == "a" or "c" or "e" or "g":
4         count += 1
```

A. 0

B. 3

C. 4

D. 6

Question 10. [1 MARK]

What does the following expression evaluate to?

```
1 ['98'] + ['76']
```

A. `'9876'`

B. `['9876']`

C. `['98', '76']`

D. `Error`

Question 11. [1 MARK]

Consider the following function.

```
1 def foo(count: int) -> bool:
2     while count < 0:
3         count += 1
4     return count < 0
```

Which option bests describes the behaviour of `foo` when invoked properly?

A. `True` *only when* count is positive.

B. `True` *only when* count is negative or zero.

C. Always `False`.

D. Always `True`.

Question 12. [1 MARK]

What does the following expression evaluate to?

```
1  7 - 4 + 3
```

A. `0`

B. `6`

C. `7`

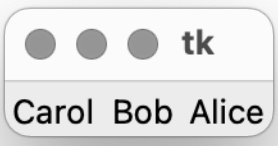
D. `Error`

Question 13. [1 MARK]

Consider the following incomplete code.

```
import tkinter as tk

window = tk.Tk()

this_side = ?

alice = tk.Label(text="Alice")
alice.pack(side=this_side)

bob = tk.Label(text="Bob")
bob.pack(side=this_side)

carol = tk.Label(text="Carol")
carol.pack(side=this_side)

window.mainloop()
```

What do we assign `this_side` in order to produce the following window?



A. `tk.LEFT`

B. `tk.RIGHT`

C. `tk.TOP`

D. `tk.BOTTOM`

Question 14. [1 MARK]

What is the value of `zs` after the following is evaluated?

```
1 ys = ['a', 'b']
2 zs = ['t']
3 ys.extend(['c'])
4 zs.append(ys)
```

A. `['t', ['a', 'b', 'c']]`

B. `['t', 'a', 'b', 'c']`

C. `['t', ['c', 'a', 'b']]`

D. `['t', 'c', 'a', 'b']`

Question 15. [1 MARK]

Which of the following statements is *True*?

A. Python will prohibit the modification of globally defined user constants like `PI = 3.14`.

B. The body of a while loop *must* execute *at least* once.

C. Every `if-then-else` statement can be written using *only* `if-then` statements.

D. The order that Python statements are given has no effect on the program's output.

Question 16. [1 MARK]

Suppose `xs` is a list. Which expression evaluates to `True` when `xs` is empty.

A. `bool(not xs)`

B. `bool(xs)`

C. `bool(len(xs))`

D. `bool(xs is [])`

Question 17. [1 MARK]

What error, if any, does Python raise when the following is executed?

```
1 def foo(x: str) -> str:
2     return 3*x
3
4 foo(2.3)
```

A. `TypeError`

B. `ValueError`

C. `NameError`

D. No error is generated.

Question 18. [1 MARK]

What is the purpose of "setter" methods as they pertain to objects?

A. They are used to change the values of private variables.

B. They are used to retrieve the values of private variables.

C. They allow private variables to be manipulated by multiple instances of the same class.

D. They are used to write data to files.

Question 19. [1 MARK]

What is the value of `ys` after the following is executed?

```
1 xs = "hello world"
2 ys = xs
3 xs[0] = "H"
4 xs[6] = "W"
```

A. `"hello world"`

B. `"Hello World"`

C. `"xs"`

D. `Error`

Question 20. [1 MARK]

How many of the following expressions evaluate to `True`?

```
1 bool("")    # empty string
2 bool(" ")   # one space
3 bool([0])
4 bool(-1)
```

A. 1

B. 2

C. 3

D. 4

Question 21. [1 MARK]

The following is a recursive function with a partially implemented base case; it concatenates a list of strings. What should we replace `#sub` with to complete this function?

```
1 def concat(xss: list[str]) -> str:
2     """
3     >>> concat(["a", "b", "c"])
4     'abc'
5     """
6     n, base = #sub
7     if len(xs) == n:
8         return base
9     return xss[0] + concat(xss[1:])
```

A. `(0, xs[0])`

B. `(1, xs[0])`

C. `(0, "")`

D. `(1, "")`

Question 22. [1 MARK]

What is the most appropriate type hint (i.e. type contract) for the following?

```
1  def foo(x, y):
2      ans = ""
3      for n in x:
4          for m in y:
5              ans += n*m
6      return ans
```

A. `foo(x: int, y: str) -> str:`

B. `foo(x: int, y: list[str]) -> str:`

C. `foo(x: list[int], y: str) -> str:`

D. `foo(x: list[int], y: list[str]) -> list[str]:`

Question 23. [1 MARK]

What error is raised after executing the following?

```
1  for k in range(10):
2      count = count + 1
3  print(count)
```

A. `TypeError`

B. `ValueError`

C. `NameError`

D. No error is generated.

Question 24. [1 MARK]

Which function *requires* the use of a a global variable to be implemented?

A. A function that calls itself.

B. A function that returns the number of times the function has been called.

C. A function that calls a different function.

D. A function that prints *and* returns a value.

Question 25. [1 MARK]

Which of the following is *not* a valid list in Python?

A. `['one', 2, '3', 'IV']`

B. `[1, int(2), [{}], 4.0]`

C. `[1, [False, True], int(2), True]`

D. All are valid lists.

Question 26. [1 MARK]

What is stored in `xs` after the following is executed?

```
xs = [1, 2, 3].reverse()
```

given that

```
>>> help(list.reverse)
reverse(self, /)
    Reverse *IN PLACE*.
(END)
```

A. `[1, 2, 3]`

B. `[3, 2, 1]`

C. `None`

D. `Error`

Question 27. [1 MARK]

What error is generated by executing the following?

```
x = int("three")
```

A. `TypeError`

B. `ValueError`

C. `NameError`

D. No error is generated.

Question 28. [1 MARK]

Suppose the following function definition has been made.

```
def foo(x: int, y: int):
    print(x/y)
```

What is `type(foo(1, 2))`?

A. `<class 'int'>`

B. `<class 'float'>`

C. `<class 'str'>`

D. `<class 'NoneType'>`

Question 29. [1 MARK]

Suppose the following lines of code have been executed.

```
1  class Artist():
2      def __init__(self, name: str, num_good_songs: int) -> None:
3          self._name = name
4          self._num_good_songs = num_good_songs
5
6  drake  = Artist("Drake", 0)
7  drizzy = Artist("Drake", 1)
```

What is stored in `drake._num_good_songs`?

A. `0`

B. `1`

C. `'Drake'`

D. `Error`

Question 30. [1 MARK]

Given the following code:

```
1  x = input(" Enter the first number: ")
2  y = input(" Enter the second number: ")
3  print(f"x + y = {x+y}")
```

and assuming the user inputs `4` then `0`. What is output?

A. `x + y = x + y`

B. `x + y = 4`

C. `x + y = 4 + 0`

D. `x + y = 40`

The following will be used to match your exam with your name. Please use BLOCK LETTERS *and write as legibly as possible.*

Student Number

Family Name

Given Name

Fill in the Blank

The next *five* questions refer to the following class definitions.

```
1  class A():
2      def __init__(self, x):
3          self.x = x
4
5      def f(self, x):
6          return self.g(x) - 1
7
8      def g(self, x) :
9          return 2*x
10
11 class B(A):
12     def g(self, y):
13         return self.x + y
14
15 class C(B):
16     def __init__(self, x, y):
17         super().__init__(x)
18         self.y = y
19
20     def f(self, x):
21         return self.x + self.y
22
23 class D(B):
24     def __init__(self, x, y):
25         super().__init__(x)
26         self.x += y
27         self.y = y
28
29     def g(self, y):
30         return self.y + y
31
32     def f(self, x):
33         return super().f(x) - x
34
35 a = A(3)
36 b = B(2)
37 c = C(2, 4)
38 d = D(1, 3)
```

Write a *single number* in the answer box *and nothing else*.

Question 31. [1 MARK]

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

Question 32. [1 MARK]

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

Question 33. [1 MARK]

What does `a.g(3)` return?

Question 34. [1 MARK]

What does `a.f(3)` return?

Question 35. [1 MARK]

What does `d.f(2)` return?

Full Solution

Question 36. [5 MARKS]

Write a function `foo` that satisfies the following specification.

```
def foo(f_path: string, word: str) -> list[str]:
        """
        The text of a book with all punctuation removed is stored at <f_path>.
        Each line of the file corresponds to one line of the book.

        Return the list of words that appear immediately before an instance of
        <word> in the file located at <f_path>.

        Preconditions:
            1.  The file at <f_path> exists.
            2.  The first word of the file is NOT <word>

        Example:
        Suppose run.txt contains the following lines:
                See spot run
                run spot see
                spot run spot
                run see spot
                see spot spot

        >>> foo("run.txt", "spot")
        ['See', 'run', 'see', 'run', 'see', 'spot']
        """
```

Write your answer on the next page.

END OF EXAMINATION