



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 Electrical Engineering & Computer Science
Semester Two 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.

Indicate your answer to the first 30 questions on the GradeScope bubble sheet.

For Examiner Use Only

Question

Mark

[illegible]

Total

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

Multiple Choice

Question 1. [1 MARK]

What is stored in `x` after only the following code is executed by Python?

```
1 Pi = 3.14
2 x = 2*'Pi'
```

- A. 6
- B. 6.0
- C. 6.18
- D. 'PiPi'

Question 2. [1 MARK]

Which statement below is *false*?

- A. We do not have to verify preconditions because it is the user's fault for breaking them.
- B. Functions have a "local scope" which means we can use the same variable-name in different functions instead of having to use many unique variable-names.
- C. A constant value in Python *can* be modified.
- D. Python prohibits the modification of private variables and the *only* way to change one is with a getter.

Question 3. [1 MARK]

What is stored in `x` after only the following code is executed by Python?

```
1 def foo(x: int, y: int) -> int:
2     if x > y:
3         z = x - y
4     elif x < y:
5         z = y - x
6     return z
7
8 x = foo(6, 6)
```

- A. 0
- B. 6
- C. -6
- D. Error

Question 4. [1 MARK]

What does the following arithmetic expression evaluate to in Python?

```
2 ** 3 % 5 - 1
```

A. 0

B. 2

C. 4

D. 7

Question 5. [1 MARK]

What is the value of `xs` after *only* the following is evaluated?

```
1 xs = (1, 2, 3)
```

```
2 ys = xs
```

```
3 ys[1] = 0
```

A. (0, 2, 3)

B. (1, 0, 3)

C. None

D. Error

Question 6. [1 MARK]

What is stored in `x` when *only* the following is executed by Python.

```
1 x = len("1\t2\t3")
```

A. 5

B. 7

C. 11

D. It depends on the number of spaces in a tab.

Question 7. [1 MARK]

What is the value of `y` after *only* the following has been executed.

```
1 y = 0
```

```
2 for x in range(5):
```

```
3     if x == 2 or 3 or 4 or 5:
```

```
4         y += 1
```

A. 2

B. 3

C. 4

D. 5

Question 8. [1 MARK]

How many of the following statements evaluate to True?

```
1 bool("False")
```

```
2 bool(" ") # 1 space
```

```
3 bool(False and True or True)
```

A. 0

B. 1

C. 2

D. 3

Question 9. [1 MARK]

What is stored in *y* after *only* the following code is executed.

```
1 def foo(xs: str) -> str:
2     if xs:
3         return 2*xs
4 y = foo("")
```

- A. "" (empty string)
- B. " " (one space)
- C. " " (two spaces)
- D. None

Question 10. [1 MARK]

What is the appropriate type-hint for the following function, assuming the function contains valid code and can be called without generating an error?

```
1 def foo(x, y, z):
2     for z in y:
3         if x in z:
4             z.append(x)
5     return x * y[0] + z
```

- A. foo(x: str, y: list[str], z: str) -> str
- B. foo(x: int, y: list[list[int]], z: list[int]) -> list[int]
- C. foo(x: int, y: list[str], z: str) -> None
- D. foo(x: int, y: list[int], z: int) -> list[int]

Question 11. [1 MARK]

Suppose we define a new class for hospital admissions. Which name is most appropriate for this class?

- A. hospital_admissions
- B. HOSPITAL_ADMISSIONS
- C. HospitalAdmissions
- D. __hospital_admissions__

Question 12. [1 MARK]

What is the value of x after *only* the following code is run?

```
1 x = 0
2
3 def foo(x: int) -> int:
4     x = x + 1
5
6 foo(x)
7 foo(x)
```

A. 0

B. 1

C. 2

D. Error.

Question 13. [1 MARK]

The following is an incomplete recursive function that counts the numbers of times a given integer occurs in a given list.

```
1 def foo(xs: list[int], x: int) -> int:
2     """
3     >>> foo([10, 10, 20], 20)
4     1
5     >>> foo([10, 10, 20], 10)
6     2
7     """
8     a, b = ?
9     if len(xs) == a:
10         return b
11     return (xs[0] == x) + foo(xs[1:])
```

A. 0, 0

B. 1, 1

C. 0, `xs[0] == x`D. 1, `xs[0] == x`

What should be assigned to a , b at line 8 to make the function work?

Recall `int(True) == 1` and `int(False) == 0`.

Question 14. [1 MARK]

What best describes the behaviour of the following function?

```
1 def foo(y, z):  
2     for x in y:  
3         if x > z:  
4             return True  
5         else:  
6             return False  
7     return False
```

- A. Always returns True.
- B. Always returns False.
- C. Returns False only when *there is no* element of y that is strictly greater than z.
- D. Returns True only when *the first* element of y is strictly greater than z.

Question 15. [1 MARK]

What is stored in x after running *only* the following code.

```
1 x = "drake is overrated".capitalize()
```

Given that

```
2 Help on method_descriptor:  
3  
4 capitalize(self, /)  
5     Return a capitalized version of the string.  
6  
7     More specifically, make the first character have  
8     upper case and the rest lower case.
```

- A. "Drake is overrated"
- B. "Drake Is Overrated"
- C. "DRAKE IS OVERRATED"
- D. None

Question 16. [1 MARK]

Suppose the following lines of Python have been executed.

```
1 letters = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
2 xs = letters[5:-5:1]
```

What is stored in xs?

- A. ['F', 'E']
- B. ['F', 'E', 'D']
- C. []
- D. Error

Question 17. [1 MARK]

What type of error is thrown by executing the following code?

```
1 def foo(x: str, y: str) -> str:
2     """
3     >>> foo("Monty", "Python")
4     'MontyPython'
5     """
6     return x + y
7
8 ans = foo(2, 1/3)
```

- A. TypeError
- B. NameError
- C. ValueError
- D. This is valid Python code.

Question 18. [1 MARK]

What is stored in x after *only* the following is executed.

```
x = (1) + (2) + (3)
```

- A. (1, 2, 3)
- B. (1, (2, 3))
- C. ((1, 2), 3)
- D. None of the above.

Question 19. [1 MARK]

What is the behaviour of the following function, supposing it is called properly (i.e. preconditions are satisfied)?

```
1 def foo(xss: list[list[int]]) -> dict[list[int], int]:
2     """ Precondition: len(xss) > 0
3     """
4     ans = dict()
5     for xs in xss:
6         ans[xs] = sum(xs)
7     return ans
```

- A. foo always returns None.
- B. foo always raises an error.
- C. foo returns a dictionary mapping lists to their sums.
- D. foo has a logical error because many lists can have the same sum.

Question 20. [1 MARK]

Which is *not* a Python naming convention/rule?

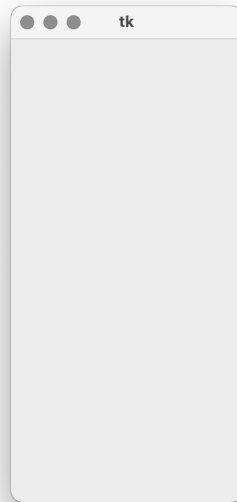
- A. Class attributes that should not be changed by the user (i.e. private variables) start with an underscore `_like_this`.
- B. Magic methods are surrounded by double underscore `__like_this__`.
- C. Global constants are capitalized `LIKE_THIS`.
- D. Class attributes are camel cased `likeThis`.

Question 21. [1 MARK]

What is the value of `xs` after executing the following?

```
1 ys = ["A", "B"]
2 xs = ["D"]
3 ys.extend(["C"])
4 xs.append(ys)
```

- A. `["D", ["A", "B", "C"]]`
- B. `["D", "A", "B", "C"]`
- C. `["D", ["A", "B", ["C"]]]`
- D. Error

Question 22. [1 MARK]

What line of code should replace #sub in order to generate the window illustrated above?

```
1 import tkinter as tk
2 root = tk.Tk()
3 #sub
4 root.mainloop()
```

- A. `root.geometry("200x400")`
- B. `root.geometry("200 x 400")`
- C. `root.geometry("400x200")`
- D. `root.geometry("400 x 200")`

Question 23. [1 MARK]

Suppose we want to assign True to the name `validate` when a user has inputted *only* the single digit '1', '2', '3', or '4' (and false otherwise). Which proposition should replace #sub to accomplish this?

```
1 value = input("Enter a single digit: ")
2 validate = #sub
```

- A. `value == "1" or "2" or "3" or "4"`
- B. `value in "1234"`
- C. `"1" <= value <= "4"`
- D. `int(value) in range(1, 5)`

Question 24. [1 MARK]

What exception should be used at <Error> to complete the function?

```
1 def foo() -> int:
2     """
3     Prompts the user to enter an integer.
4     Repeats until a number is entered.
5     """
6     try:
7         return int(input("Enter an integer "))
8     except <Error>:
9         return foo()
```

- A. TypeError
- B. ValueError
- C. InputError
- D. IntError

Question 25. [1 MARK]

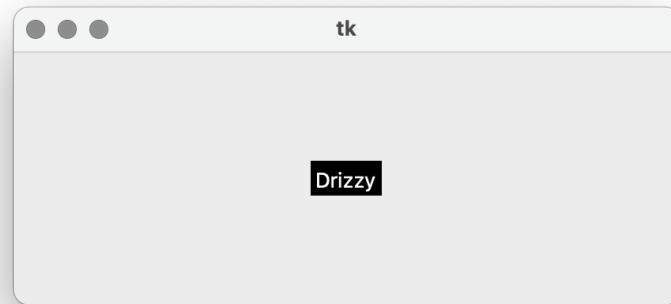
Consider the following function.

```
1 def foo(xs: list[int], ys: dict) -> bool:
2     for x in xs:
3         if not x in [ys[k] for k in ys]:
4             return False
5     return True
```

- A. foo always returns True.
- B. foo always returns False.
- C. foo returns True only when every element of xs is a *key* of ys.
- D. foo returns True only when every element of xs is a *value* of ys.

Question 26. [1 MARK]

The following tkinter window has been stretched using the mouse.



What line of code should replace #sub in order to generate the window illustrated above?

```
1 import tkinter as tk
2 root = tk.Tk()
3 label = tk.Label(text = "Drizzy", background = "black", foreground = "white")
4 # sub
5 label.pack(expand = opts[0], fill = opts[1])
6 root.mainloop()
```

A. `opts = Tk.TRUE, Tk.NONE`

B. `opts = Tk.FALSE, Tk.BOTH`

C. `opts = Tk.NONE, Tk.TRUE`

D. `opts = Tk.BOTH, Tk.FALSE`

Question 27. [1 MARK]

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

A. They are used to change the value of a private variable.

B. They are used to retrieve the value of a private variable.

C. They change a private variable to a public one and vice-versa.

D. They ensure that all private variables have the correct type.

Question 28. [1 MARK]

How many exclamation marks (!) are in `output.txt` after running (only) the following code?

```
1 the_file = open("output.txt", 'a')
2 for k in range(1, 4):
3     the_file.write(k * '!')
4 the_file.close()
```

A. 1

B. 3

C. 6

D. Impossible to deduce without knowing the initial contents of `output.txt`.

Question 29. [1 MARK]

Which of the following is *not* a necessary feature of an imperative programming language?

A. Iteration.

B. Selection.

C. Transparency.

D. Sequencing.

Question 30. [1 MARK]

What is the value of `z` after running the following code?

```
1 xs = ['a', (3,4), {1: 'b'}]
2 ys = xs.copy()
3 ys[2] = {2: 'c'}
4 z = xs[2][1]
```

A. 'a'

B. 'b'

C. 'c'

D. Error.

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     def f(self, x):
5         return self.g(x) - 1
6     def g(self, x):
7         return 3 * self._x
8     def fg(self, x):
9         return self.f(x) * self.g(x)
10
11 class B(A):
12     def f(self, y):
13         return self._x - y
14     def g(self, y):
15         self._y = y
16         return self._x - self._y
17
18 class C(B):
19     def __init__(self, x, y):
20         super().__init__(x)
21         self._y = y
22         self._x += y
23     def f(self, x):
24         return A.g(self, x) + self._y
25
26 class D(C):
27     def __init__(self, x, y):
28         self._x = x
29         self._y = y + self._x
30     def g(self, y):
31         return super().g(y)
32     def fg(self, y):
33         return self._x * self._y
34
35 a = A(0)
36 b = B(1)
37 c = C(1,1)
38 d = D(1,1)
```

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

Question 31. [1 MARK]

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

Question 32. [1 MARK]

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

Question 33. [1 MARK]

What does `c.fg(2)` return?

Question 34. [1 MARK]

What does `d.fg(1)` return?

Question 35. [1 MARK]

What does `d.g(4)` return?

Full Solution

Question 36. [5 MARKS]

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

```
1 def substring(xs: str, ys: str) -> bool:
2     """
3     The SUBSTRINGS of ys are all strings arrived at by deleting zero
4     or more elements of ys. For example the substrings of "abc" are
5     "abc", "ab", "ac", "bc", "a", "b", "c" and "" (empty string).
6
7     Return True when xs is a substring of ys.
8
9     >>> substring("ac", "abc")
10    True
11    >>> substring("ca", "abc")
12    False
13    """
```

Write your answer on the next page.

Write your answer on the next page.

Write your answer on the next page.

END OF EXAMINATION