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, 2024

CSSE7030 Introduction to Software Engineering

This paper is for St Lucia Campus students.

Examination Duration: 120 minutes

Planning Time: 10 minutes

Exam Conditions:

•No written or printed material 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

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.
Answer questions 31 to 36 in the spaces provided.

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]

What does the following arithmetic expression evaluate to in Python?

```
2 ** 2 % 4 - 2
```

A. `2`
B. `-2`
C. `1`
D. `-1`
E. None of the above

Question 2. [1 MARK]

What is stored in x when only the following is executed by Python ?

```
x = len("\n\t23\t\n")
```

A. `10`
B. `6`
C. `2`
D. It depends on the number of spaces in a tab.
E. None of the above

Question 3. [1 MARK]

After starting up the Python interpreter, the following code (and only the following code) is entered.

```
if True or x:
    x = 1
else:
    x = 0
```

What error, if any, does this code raise?

A. `TypeError`
B. `ValueError`
C. `NameError`
D. `SyntaxError`
E. This is valid Python code.

Question 4. [1 mark]

What is the value of x after only the following code is executed?

```
x = 'Hello World' - 2 * 'Hello World'
```

A. `''` (the empty string)
B. `Hello WorldHello World`
C. `Hello World`
D. `Error`
E. None of the above.

Question 5. [1 mark]

What is stored in y after only the following is entered into Python?

```
y = ':'.join('hands \t many'.split('\t'))
```

A. `hands:many`
B. `hands : many`
C. `hands: many`
D. `Error`
E. None of the above

Question 6. [1 mark]

The following is a recursive function with a partially implemented base case; it counts the number of elements in a list. What should we replace `#sub` with to complete this function?

```
def count(xs: list[int]) -> int:
    """
    >>> count([10, 20, 30])
    3
    """
    (a, b) = #sub

    if len(xs) == a:
        return b

    return 1 + count(xs[1:])
```

A. `(1, 1)`
B. `(0, 0)`
C. `(0, 1)`
D. `(1, 0)`
E. None of the above

Question 7. [1 mark]

Given the following code:

```
x = input("Prompt: ")
y = input("Prompt: ")
print(f"x + y = {int(x + y)}")
```

If the user types 5 at the first prompt then 2 at the second prompt, what is printed?

A. `x + y = 7`

B. `x + y = '7'`

C. `x + y = 52`

D. `x + y = '52'`

E. `Error`

Question 8. [1 mark]

What is stored in `y` after only the following code is executed?

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

y = ['a', 'b']
g(y, 'c').append(g(y.copy(), 'c'))
```

A. `['a', 'b', 'c']`

B. `['a', 'b', 'c', 'c']`

C. `['a', 'b', 'c', 'a', 'b', 'c']`

D. `['a', 'b', 'c', ['a', 'b', 'c', 'c']]`

E. `Error`

Question 9. [1 mark]

Suppose some code has been styled in accordance with the style guide used in this course. What can be deduced about the name `FOO_BAR`?

`FOO_BAR` is a ...

A. class name

B. constant variable

C. method

D. private variable

E. None of the above

Question 10. [1 MARK]

Suppose the following function definition has been made.

```
def foo(x, y):
    if x - y == 0:
        return x

    return foo(x-1, y)
```

What does `foo(-1, 1)` return?

A. `0`
B. `1`
C. `-1`
D. `-2`
E. Error

Question 11. [1 MARK]

What is the value of `x` after only the following has been evaluated?

```
x = "Hello Hello".find("Hello")
```

Given that:

```
S.find(sub[, start[, end]]) -> int

    Return the lowest index in S where substring sub is found,
    such that sub is contained within S[start:end].  Optional
    arguments start and end are interpreted as in slice notation.

    Return -1 on failure.
```

A. `0`
B. `6`
C. `-1`
D. `None`
E. `ValueError`

Question 12. [1 MARK]

What exception should be used to replace `<Error>` at line 8 to complete the function according to specification?

```
1 def get_element(xs: list[int], index: int) -> int:
2     """
3     Retrieves the element at the provided index in the list.
4     Continues prompting the user until a valid index is entered.
5     """
6     try:
7         return xs[index]
8     except <Error>:
9         return get_element(xs, int(input("Enter a valid index: ")))
```

A. `TypeError`

B. `NameError`

C. `IndexError`

D. `KeyError`

E. `ValueError`

Question 13. [1 MARK]

What is the value of `b` after the following code is executed?

```
def f(x):
    a = 5
    x = x / a
    return a+x

a = 10
b = f(a)
```

A. `7`

B. `7.0`

C. `15.0`

D. `15`

E. Error

Question 14. [1 MARK]

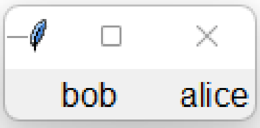
What replaces `#sub1` and `#sub2` in the following code to generate the image to its right?

```python
import tkinter as tk
root = tk.Tk()
s1 = #sub1
s2 = #sub2

alice = tk.Label(root, text="alice")
alice.pack(side = s1[0], expand = s1[1])

bob = tk.Label(root, text="bob")
bob.pack(side = s2[0], expand = s2[1])

root.mainloop()
```



A. `sub1 = (tk.RIGHT, tk.FALSE), sub2 = (tk.LEFT, tk.TRUE)`
B. `sub1 = (tk.RIGHT, tk.TRUE), sub2 = (tk.LEFT, tk.FALSE)`
C. `sub1 = (tk.LEFT, tk.FALSE), sub2 = (tk.RIGHT, tk.TRUE)`
D. `sub1 = (tk.LEFT, tk.TRUE), sub2 = (tk.RIGHT, tk.FALSE)`
E. None of the above.

Question 15. [1 MARK]

What is the value of `y` after the following statements are evaluated?

```python
x = ['hello', 'HELLO', 'world', 'WORLD']
y = x[-1][-5]
```

A. `'d'`
B. `'D'`
C. `'w'`
D. `'W'`
E. Error

Question 16. [1 MARK]

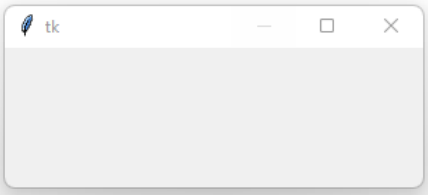
Consider the function `product` defined below that multiplies two numbers. What is the value of `x`?

```python
def product(num1: int, num2: int):
    print(2 * num1)
    return num1 * num2
    return

x = product(3, 4.0)
```

A. `6`
B. `12.0`
C. `12`
D. `None`
E. Error

Question 17. [1 mark]



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

```
import tkinter as tk
root = tk.Tk()
#sub
root.mainloop()
```

A. `root.geometry("300x100")`
B. `root.geometry(300x100)`
C. `root.geometry("100x300")`
D. `root.geometry(100x300)`
E. None of the above.

Question 18. [1 mark]

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

A. They are used to retrieve the value of a private variable.
B. They are used to change 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.
E. None of the above.

Question 19. [1 mark]

What is stored in x after only the following is entered into Python?

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

A. `(1, 2, 3)`
B. `(3, 5)`
C. `(1, 2, 2, 3)`
D. `[(1, 2), (2, 3)]`
E. Error

Question 20. [1 MARK]

What error (if any) will the following code produce when executed by Python?

```
def concatenate(xs: list[int], ys: list[int]) -> list[int]:
    return xs + ys

concatenate(' ', '2a')
```

A. `SyntaxError`
B. `TypeError`
C. `NameError`
D. `ValueError`
E. This is valid Python code.

Question 21. [1 MARK]

What is the value of `ys` after only the following has been evaluated?

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

A. `[3, 4, 5]`
B. `[9, 16, 25]`
C. `[3, 4]`
D. `[9, 16]`
E. Error

Question 22. [1 MARK]

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

```
cs = 'abc'
for k, c in enumerate(cs):
    x = 2*k + c
```

A. `'abbcccc'`
B. `'abcd'`
C. `'aabbbbcccccccc'`
D. `Error`
E. None of the above

Question 23. [1 MARK]

After the assignment `s1 = "Hello World"` which of the following statements assigns `"o W"` to `s2`?

A. `s2 = s1[4:7]`
B. `s2 = s1[4:-4]`
C. `s2 = s1[-7:-4]`
D. All of the above
E. None of the above

Question 24. [1 MARK]

Consider the following assignments.

```
xss = ['abcdef', 'ABCDEF', 'ghijkl']
yss = ['def', 'DEF']
```

How many of the following expressions are equivalent to yss?

```
[xs[-3:] for xs in xss[0:2]]
[xs[3:] for xs in xss[0:-1]]
[xs[-3:] for xs in xss[-3:-1]]
[xs[3:] for xs in xss[-3:2]]
```

A. 0
B. 1
C. 2
D. 3
E. 4

Question 25. [1 MARK]

How many stars (*) are in output.txt after calling foo without generating an error?

```
def foo() -> None:
    xs = ['*', '**', '***']
    for x in xs:
        with open("output.txt", "w") as f:
            f.write(2*x)
    return
```

A. 0
B. 3
C. 6
D. 12
E. Impossible to deduce without knowing the initial contents of output.txt.

Question 26. [1 MARK]

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

```
def foo(xs: list[str]) -> list[str]:
    if xs:
        xs.append(" ")
        return xs
    return []

y = foo([""])
```

A. ["", " "]
B. [" "]
C. [""]
D. []
E. Error

Question 27. [1 mark]

What is the best description of the behaviour of the following function?

```
def bar(d1: dict, d2: dict) -> bool:
        for x in d1:
              if x not in d2 or d1[x] != d2[x]:
                   return False
        return True

```

A. bar only returns `True` when all keys in `d2` exist in `d1` and `False` otherwise.

B. bar only returns `True` when all key-value pairs in `d1` exist in `d2` and `False` otherwise.

C. bar always returns `False`.

D. bar always returns `True`.

E. bar always throws errors

Question 28. [1 mark]

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

```
xs = [['12'], {1: '1'}]
xs[1] = {xs[0] : '2'}
```

A. `[['12'], {['12']: '2'}]`

B. `[['12'], {['1']: '2'}]`

C. `[['12'],{'12': '2'}`

D. `Error`

E. None of the above.

Question 29. [1 mark]

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

```
xs = 'hello'
ys = xs
ys[0] = 'H'
```

A. `'hello'`

B. `'Hello'`

C. `'H'`

D. `Error`

E. None of the above

Question 30. [1 MARK]

Which statement is false?

A. Type-hints are not enforced by Python.

B. Python prohibits the user from changing constant variables.

C. Functions can be defined inside of functions.

D. Every for loop can be written as a while loop.

E. None of the above.

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.

```
class A(object) :
    def __init__(self, x) :
        self._x = 2*x

    def f(self, x) :
        return x + self._x

    def g(self, x) :
        return 2 * self.f(x) - x

class B(A) :
    def f(self, x) :
        return self._x - x

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

class D(B) :
    def __init__(self, x, y) :
        super().__init__(x)
        self._x += 2*y
        self._y = y

    def f(self, x) :
        return self._y + x

    def g(self, x) :
        return super().g(x) - x

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

Question 31. [1 MARK]

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

Question 32. [1 MARK]

What does `b.g(1)` return?

Question 33. [1 MARK]

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

Question 34. [1 MARK]

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

Question 35. [1 MARK]

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

Full solution

Question 36. [5 MARKS]

Implement the following function according to its specification. Do not include a docstring.

```
def remove_adjacent_pairs(cs: str) -> str:
    """
    Given a string cs, return the string obtained after removing all adjacent pairs of
    duplicate characters from cs. This process should be repeated until no adjacent
    duplicates remain.

    Parameters:
        cs: A string that needs to be processed.

    Return:
        A modified version of cs where all adjacent pairs of duplicate characters have
        been removed.

    Examples:
        >>> remove_adjacent_pairs("abbaca")
        'ca'
    because "abbaca" -> "aaca" -> "ca" after removing adjacent pairs.

        >>> remove_adjacent_pairs("aaac")
        'ac'
     because "aaac" ->  "ac" after removing one pair of "a"s.


        >>> remove_adjacent_pairs("azxxzy")
        'ay'
    because "azxxzy" -> "azzy" -> "ay" after removing adjacent pairs.

        >>> remove_adjacent_pairs("aabbcc")
        ''
    because "aabbcc" -> "" after removing all adjacent pairs.

    Note that after each removal of adjacent duplicated pairs, the process continues
    on the resulting string until no more adjacent duplicated pairs exist.
    """
```

Write your answer on the next page.

Write your answer here:

END OF EXAMINATION