CSCI 1470 CS 1
Fall 2025
Final Examination

Last Name: _____ **First Name:** _____ **Student Id:** _____

Number: _____

Time allowed: *2 hours.* Total score: 100 points. *Closed* book examination. *Two letter-size information sheets (both sides) prepared by yourself are allowed.* This question paper is printed on *both sides*.

Answer all questions. Turn in everything: question and answer papers, information sheets and sketch papers. They will be stapled together.

[Q1] (24%) Multiple Choices. Please circle a choice as your answer. Alternatively, you can write your choice clearly after the question.

[1] What value will be printed?

```
my_list = [3, -4, 0, -1, 2, 1, 8]
n = 0
count = 0

while n < len(my_list):
    if my_list[n] > 0:
        count = count + 1
    n = n + 1
print(count)
```

A. 1
B. 3
C. 4
D. 5

[2] What is the output?

```
x = 18
while x % 3 == 0:
    print(x, end=" ")
    x = x // 3
```

A. 6
B. 6 2
C. 18 6 2
D. 18 6

[3] Which of the following loops is best implemented with a for loop?

A. Asking a user to enter names until the user enters "Quit".
B. Counting the number of negative values in a list of integers.
C. Starting from a user-entered integer, increment the value until the value is a prime number.
D. Reading values from a temperature sensor until it gives a value greater than 100 degrees.

[4] What is the output of the following code?

```
z = 0
a = 5
while a > 0:
    a = a - 1
    if a == 2:
        continue
    z = z + a
print(z)
```

A. 7
B. 8
C. 9
D. 10

[5] A programmer must write a 500 line program. Which is most likely the best approach?

A. Write 1 line, run and debug, write 1 more line, run and debug, repeat
B. Write 10-20 lines, run and debug, write 10-20 more lines, run and debug, repeat
C. Write 250 lines, run and debug, write 250 lines, run and debug
D. Write 500 lines, run and debug

[6] What is output?

```
def calc(num1, num2):
    return 1 + num1 + num2
print(calc(4, calc(1, 2)))
```

A. 7
B. 8
C. 9
D. 10

[7] What is the result when the program is executed?

```
def add(x, y):
    return x + y
print("Begin test")
s = add("hello", 5)
print(s)
print("End test")
```

A. The program outputs "Begin test", then an error is generated, and the program exits.
B. An error is generated before anything is printed.
C. The program outputs "Begin test", followed by "hello5", followed by "End test"
D. The program outputs "Begin test", followed by "End test", and no other text is printed.

[8] Which variable can be used in place of XXX in the following code?

```
def fahrenheit_to_celsius(f):
    fraction = 5 / 9
    c = (f - 32) * fraction
    return c
degrees = float(input("Enter degrees in Fahrenheit: "))
print(fahrenheit_to_celsius(XXX))
```

A. degrees
B. f
C. c
D. fraction

[9] What will be output for `print(f"{1.6180339:.5f}")` ?

A. 1.61803
B. 00001.61803
C. 00001.6180
D. 1.6180

[10] What is the output?

```
new_list = ["python", "development"]
new_list.append("in progress")
print(new_list)
```

A. ['python', 'development', 'in progress']
B. ['python', 'development', ['in progress']]
C. ['python', 'in progress']
D. ['python', 'developmentin progress']

[

[11] What is the output?

```
b1 = [7, 5, 9, 6]
b1 = sorted(b1)
b2 = b1
b2.append(2)
print(b1, b2)
```

A. [5, 6, 7, 9, 5, 6, 7, 9, 2]
B. [5, 6, 7, 9, 2] [5, 6, 7, 9, 2]
C. [2, 5, 6, 7, 9] [2, 5, 6, 7, 9]
D. [5, 6, 7, 9] [5, 6, 7, 9, 2]

[12] How do you obtain a list of tuples of key-value pairs in a dictionary dict?

A. dict.get(key, default)
B. dict.values()
C. dict.items()
D. dict.keys()

[Q2] (15%) True or False (Circle one choice, or write either True or False)

[a] [T F] The continue statement in Python is a control flow statement used within a loop to exit the loop.

[b] [T F] In Python, a function is an object.

[c] [T F] In Python, FileNotFoundError may be raised when executing the statement `open("my_file.txt")`.

[d] [T F] In Python, the file mode `"a"` is used for asynchronously writing to a file.

[e] [T F] In Python, the key of a dict can be a string.

[f] [T F] In Python, the print() method can accept multiple number of arguments.

(g) [T F] The sys module is a built-in module in Python.

(h) [T F] The try-except statements in Python handles syntax errors.

(i) [T F] In Python, `"abc" >= "ab"`.

(j) [T F] In Python, `"123" >= "2"`.

[Q3] (12%) Strings, lists and slices

[a] (4%) What is the output of executing the following code?

```
s = "abcdefg"
print(s[1])
print(s[1:3])
print(s[0:6:2])
print(s[::-1])
```

[b] (8%) What is the output of executing the following code?

```
list_1 = [1, 2, [3, 4], 5]
print(list_1)
print(list_1[1])
print(list_1[2][1])
print(list_1[1:3])
list_2 = list_1
list_1[0] = 9
list_1[2].append(20)
print(list_2[0])
print(list_2[2])
list_3 = list_1[:]
list_1[0] = 11
list_1[2].append(33)
print(list_3[0])
print(list_3[2])
```

[Q4] (30%)

[1] (8%) Write Python code using the range function to print the squares of the odd integers from 1 to 11 in the following manner.

```
1 * 1: 1
3 * 3: 9
5 * 5: 25
7 * 7: 49
9 * 9: 81
11 * 11: 121
```

[2] (8%) Write Python code to count and print the total number of characters in a list of string, `list_1`. For the following content of list_1, your code should output 19.

```
list_1 = ["hello", "world", "It is me."]
```

[3] (7%) Write python code to count and print the number of strings in a list of string, list_1, that starts with the two alphabets "al". For the following list_1, your code should print 3. Tips: the string method s.startswith(prefix) returns true if the string s starts with the prefix string.

```
#   "all in one".startswith("all") returns true.
#   "all in one".startswith("att") returns false.

list_1 = ["mouse", "a2", "all", "alright", "allow", "dog"]
```

[4] (7%) Consider a dictionary, d, with strings as keys and integers as values. Write Python code to sum all the integer values in d with keys that start with an "a". For the following dictionary d, your code should output 45.

```
d = {
    "mouse": 4,
    "a1": 10,
    "a2": 20,
    "a cat": 15,
    "dog": 22
}
```

[Q5] (20%) functions

[1] Write a function make_dict to accept a list of strings and return a dict with the items of the list as its key. The values are the position of the items in the list, starting from 1. For example, executing the following code:

```
list_1 = ['cat', 'dog', 'tiger', 'lion']
print(make_dict(list_1))
```

output:

```
{'cat': 1, 'dog': 2, 'tiger': 3, 'lion': 4}
```

[2] Write a function to accept a list of lists of integers and return the sum of all integers in the parameter. No input error checking is required. For example, executing the following code,

```
List_1 = [[1,2,3], [2,3], [4], [2,1,3,4]]
print(f"sum_list_integer(list_1): {sum_list_integer(list_1)}")
```

output:

```
sum_list_integer(list_1): 25
```