ANSWER SHEET

DECEMBER 2016

INSTRUCTOR

DATE
Dec 18, 2023

DO NOT USE PHOTOCOPIED ANSWER SHEETS AS THEY DO NOT SCAN ACCURATELY.

MARKING INSTRUCTIONS

DIRECTIONS
Read each question and its lettered answer. When you have decided which answer is correct, completely blacken the corresponding bubble. Make no stray marks; they may count against you.

SAMPLE

1. CALGARY is
   a. a country
   b. a mountain
   c. an island
   d. a city
   e. a province

(A) (B) (C) ● (E)

STUDENT NUMBER: 0 0 1 1 2 2 3 3

If your Student ID Number starts with "00" please include the leading 0's e.g. 00123456

COURSE NAME: C P S C

COURSE NUMBER: 2 3 1

VER. (Exam Version): 0

SECT. (Lecture Section):

LAST NAME: P Y T H O N

INITIAL: 1ST M, 2ND

	T	F
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		

ADDITIONAL ANSWERS ON BACK

CPSC 231 Fall 2023

Practice Final Exam

Your Name: _____

Your Student Number: _____

Your Signature: _____

Also fill in the contact information on the scantron sheet:

Instructor: **N. Verwaal** or **A. Motamedi** or **F. Anzum**

Date: **Dec 18, 2023**

Student number: **your student number**

Course name: **CPSC**

Course number: **231**

VER. (Exam Version)**: 01 or 02**

SECT. (Lecture Section): **01** or **02** or **03**

Last name: **your last name/surname**

Initial: **the first letter of your first name**

Do not open the exam booklet until you are told to do so.
Good luck!

Duration: 2 hours

This is a closed book exam: Exams must be completed individually, following regulations on academic integrity. Only a pencil and an eraser are allowed during the exams. Electronic devices, including laptops, cell phones, and calculators, must be turned off and stored.

There are 50 multiple choice questions in total. Fill in the best answer to each question on the scantron sheet. Only one bubble per question should be filled, or the machine will not recognize your answer.

1. What Linux command could be used to update the current working directory to a different directory?
    a. ls
    b. cd
    c. mv
    d. move
    e. None of the Above

2. What Linux command could be used to list the files in the current working directory?
    a. ls
    b. cd
    c. mv
    d. move
    e. None of the Above

3. What is the type of the variable *a* when you have *a* = 10.0 ?
    a. int
    b. bool
    c. float
    d. decimal
    e. None of the Above

4. Which of the following is a data type in Python that requires an import?
    a. int
    b. float
    c. str
    d. array
    e. None of the above

5. What is the correct syntax for defining a list variable in Python?
    a. my_list = [1, 2, 3, 4, 5]
    b. my_list = list(1, 2, 3, 4, 5)
    c. list my_list = [1, 2, 3, 4, 5]
    d. [1, 2, 3, 4, 5] = my_list
    e. None of the Above

6. What type of error does the Python interpreter give when running the following code:

```
i = 1
    if i <= 2:
        print(i)
```

    a. syntax error
    b. runtime error
    c. logic error
    d. More than one error
    e. No errors

7. What is the output of the following code: print("Triangle" == "triangle")
    a. Triangle
    b. triangle
    c. True
    d. False
    e. None of the Above

8. What is the output of the following code:

```
print("Result is %.2f" % 3.1415)
```

    a. Result is %.2d
    b. Result is %.2d 3.1415
    c. Result is 3.14
    d. Result is 03
    e. None of the Above

9. What is the output of the following code:

```
a = 1
b = 2
print(a or b)
```

   a. 1
   b. 2
   c. True
   d. Python gives an error
   e. None of the Above

10. What is the output of the following code:

```
print(1=2)
```

   a. 1
   b. True
   c. False
   d. Python gives an error
   e. None of the Above

11. What is the output of the following code:

```
a = "a"
b = "b"
if a == b:
       print("Hello")
```

   a. Hello
   b. "Hello"
   c. code prints nothing
   d. Python gives an error
   e. None of the Above

12. Determine how many times the loop in the following code will run:

```
items = "###P###"
for i in items:
    if i == "P":
       break
    print(i)
```

   a. 2 times
   b. 3 times

c. 4 times
d. infinite times
e. None of the Above

13. What is the output of the following code:

```
x = 5
y = 9

if x and 10 > y:
    print("x is greater than y")
elif x and 10 < y:
    print("x is less than y")
else:
    print("x is equal to y")
```

a. x is greater than y
b. x is less than y
c. x is equal to y
d. Error in Python
e. None of the Above

14. Which of the following lines on strings is NOT correct syntax:

a. new_string = "new_string"
b. new_string = ""
c. new_string = 'new_string'
d. new_string = "new string"[0:2]
e. All of the above are correct syntax

15. Which of the following lines on strings will produce an error, given:
`new_string = "new string"`

a. another_string = new_string
b. another_string = new_string[0]
c. new_string = new_string[0]
d. new_string[0] = "N"
e. All of the above are correct

16. What is the base 10 number 45 in binary?
    a. 45
    b. 1011 0100
    c. 0011 1101
    d. 0010 1101
    e. None of the Above

17. What is the base 10 number 44 in hexadecimal?
    a. 2D
    b. 45
    c. 213
    d. 0x45
    e. None of the Above

18. What is the output of the following code:

```
num = 10
if num >= 5:
    print (1)
    if num >= 4:
        print (2)
    elif num >= 3:
        print (3)
    if num < 4:
        print (4)
    else:
        print (3)
else:
    print (5)
```

    a.
```
1
2
3
```

    b.
```
1 2 3
```

    c.
```
1
```

    d. Python gives an error
    e. None of the Above

19. Which of the following is NOT correct?
    a. You can swap any element of a list with a new element
    b. You can swap any element of a tuple with a new element
    c. You can swap any element in an array with a new element
    d. You can swap any value in a dictionary with a new value
    e. None of the above

20. What is a correct way to create a dictionary?
    a. `my_dict = ["key1": "value1", "key2": "value2"]`
    b. `my_dict = {"key1": "value1", "key2": "value2"}`
    c. `my_dict = {"key1": "value1"; "key2": "value2"}`
    d. `my_dict = ("key1": "value1"; "key2": "value2")`
    e. None of the above

21. What is a correct way to create a 2D list of size 10 by 10?
    a. `my_list = [i for i in [j for j in range(10)]]`
    b. `my_list = [0 for i in range([j for j in range(10)])]`
    c. `my_list = [[0 for i in range(10)] for j in range(10)]`
    d. `my_list = [i for i in [j for j in 0]]`
    e. None of the above

22. What is the output of running the following code:

```
a = "b"
b = "a"
a, b = b, a
print(b, a)
```

    a. a b
    b. b a
    c. a a
    d. b b
    e. None of the above

23. What is a correct way to define a custom function in Python?
    a. def my_function():
    b. my_function()
    c. new = my_function()
    d. my_function:
    e. None of the above

24. What is a correct way to call a custom function and save the return value in a variable?
    a. def my_function():
    b. my_function()
    c. new = my_function()
    d. my_function:
    e. None of the above

25. What is the output of running the following code:

```
def my_function(num=5):
    print(num)

num = 1
num += 2
my_function(num)
```

    a. 1
    b. 2
    c. 3
    d. 5
    e. None of the above

26. What is the output of running the following code:

```
def my_function(num=5):
    print(num)

num = 1
num += 2
my_function()
```

    a. 1
    b. 2
    c. 3
    d. 5
    e. None of the above

27. What is the output of running the following code:

```
import numpy as np
def my_function(num1, num2):
    x[0] = 4
    return x + y

x = np.array([1, 2, 3])
y = np.array([3, 4, 5])
print(my_function(x, y))
```

   a. [1 2 3]
   b. [4 6 8]
   c. [4 2 3 3 4 5]
   d. [7 6 8]
   e. None of the above

28. What is the output of running the following code:

```
import numpy as np
def my_function(num1, num2):
    return num1 * 2

x = np.array([1, 2, 3])
y = np.array([3, 4, 5])
print(my_function(x, y))
```

   a. [1 2 3 3 4 5]
   b. [1 2 3 1 2 3]
   c. [2 4 6]
   d. [4 6 8]
   e. None of the above

29. What is the output of running the following code:

```
import numpy as np
def my_function(num1, num2=None):
    if num2 is None:
        num1 = 'a'
    return num1

print(my_function(np.array(["a", "b", "c"])))
```

   a. a
   b. ['a' 'b' 'c']
   c. 'a'
   d. Error in Python
   e. None of the above

30. What is the output of running the following code:

```
import numpy as np
def my_function(num1, num2=None):
    if num2 is None:
        num1 = 'a'
    return num1

print(my_function(np.array(["a", "b", "c"]), ""))
```

a. a
b. ['a' 'b' 'c']
c. 'a'
d. Error in Python
e. None of the above

31. Which of the following Python statements will result in the output: 6?

```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

a. `print(A[0][0])`
b. `print(A[0][1])`
c. `print(A[1][1])`
d. `print(A[1][2])`
e. None of the above

32. Which of the following is temporary storage?
    a. USB key
    b. hard dive
    c. DVD
    d. memory
    e. The above are all temporary storage devices

33. How do you run a Python file with a command line argument under the Linux command line?
    a. python filename.py argument
    b. filename.py argument
    c. python filename.py
    d. python
    e. None of the above

34. How do you obtain the values of command line arguments using Python code?
    a. It's in the variable sys.argv
    b. It's in the variable sys.args
    c. It's in the variable sys.arg
    d. It's in the variable sys.argc
    e. None of the above

35. See the following code. What is a reason we don't use if/else instead of try/except to ensure we don't open the file if the file does not exist?

```
try:
    text_handler = open("text")
except FileNotFoundError as err:
    # do something
```

    a. The file can be changed/removed right after the code got past "if"
    b. There is no condition you can use in the "if"
    c. try/except runs faster than if/else
    d. Using if/else is more likely to cause other mistakes in the code. Just let try/except handle everything.
    e. None of the above

36. Which of the following are accurate descriptions of classes and objects in Python?
    a. You can have multiple methods in a class
    b. You can have a class inherit from another class
    c. You can have multiple objects created from a class
    d. All of the above
    e. None of the above

37. What is the output of the following code?

```
class guard:
    def __init__(self, name, id):
        self.name = name
        self.id = id
    def get_id(self):
        return self.id
    def set_id(self, new_id):
        self.id = new_id

Adam = guard("Adam", 1)
Bob = guard("Bob", 1)
```

```
print(Adam.get_id() == Bob.get_id())
```

a. True
b. False
c. 1
d. Python gives an error
e. None of the above

38. What is the output of the following code?

```
class guard:
    def __init__(self, name, id):
        self.name = name
        self.id = id
    def get_id(self):
        return self.id
    def set_id(self, new_id):
        self.id = new_id

Adam = guard("Adam", 1)
Bob = Adam
Bob.set_id(2)
print(Adam.get_id() == Bob.get_id())
```

a. True
b. False
c. 1
d. Python gives an error
e. None of the above

39. Continuing from the question above, based on this code, are Adam and Bob the same guard (same object)?
    a. Yes
    b. No
    c. Sometimes
    d. How can they be the same!?
    e. None of the above

40. The following code does not produce an error, but we shouldn't do this anyway. Why?

```
class guard:
    def __init__(self, name, id):
        self.name = name
        self.id = id
    def get_id(self):
        return self.id
    def set_id(self, new_id):
        self.id = new_id

Adam = guard("Adam", 1)
Bob = guard("Bob", 1)
print(Adam.id == Bob.id)
```

a. Adam.get_id() is better looking code than Adam.id
b. Adam.get_id() runs faster than Adam.id
c. Using a variable directly from the guard class violates the principle of encapsulation
d. Adam.id should be called self.id and this could cause unnecessary errors in the future
e. None of the above

41. How many lines are printed in the output of this code:

```
file = open("text.txt", "r")
i = 0
for line in file:
    for item in line:
        print(i, item)
        i += 1
file.close()
```

Content of text.txt is one single line with a new line after the last #, as shown below:

```
##P#

```

a. 3 lines
b. 4 lines
c. 5 lines
d. 6 lines
e. None of the above

42. Continuing from the question above, if there is no new line right after the last #, but there are 5 blank spaces after the last #, and a new line after that, how many lines are printed in the output of the same code?

```
##P#
```

a. 3 lines
b. 4 lines
c. 5 lines
d. 6 lines
e. None of the above

43. Continuing from the question above, if you want to output only the position of P (wherever the P might appear in the line, what do you write in the loop?

a.
```
if item("P"):
    print(i)
```
b.
```
if item[i]:
    print(i)
```
c.
```
if i == "P":
    print(i)
```
d.
```
if item == "P":
    print(i)
```
e. None of the above

44. What is the output of the following code?

```
def my_function(n):
    if n == 0:
        return 1
    return my_function(n-1)

print(my_function(5))
```

a. 5
b. 3
c. 2
d. 1
e. None of the above

45. What is the output of the following code?

```
def my_function(n):
    if n == 0:
        return 1
    return my_function(n-2)

print(my_function(10))
```

a. 5
b. 3
c. 2
d. 1
e. None of the above

46. What is the output of the following code?

```
def my_function(n):
    if n == 0:
        return 1
    return my_function(n-2)

print(my_function(9))
```

a. 5
b. 3
c. 2
d. 1
e. None of the above

47. What is the output of the following code?

```
def my_function(n):
    if n == 0 or n == 1:
        return 1
    else:
        return 2 * my_function(n-2)

print(my_function(9))
```

a. 5
b. 3

c. 2
d. 1
e. None of the above

48. How many errors are in the following code (only count errors that will be given by the Python interpreter)?

```
def my_function(n):
    if n < 2:
        return n
    else:
        return 2 * my_function(n)

print(my_function(5))
```

a. 5
b. 3
c. 2
d. 1
e. No errors given by Python

49. How many errors are in the following code (only count errors that will be given by the Python interpreter)?

```
def my_function(num1 = 1, num2, num3 = 2):
    print(num1)
    print(num2, num3)
    print(num2 + num3)

my_function(num2 = 1)
```

a. 1
b. 2
c. 3
d. 4
e. More than 4

50. We want to write a recursive function that calculates the sum of all elements of a list. The following are pieces of the code:

(1) def sum_list(lst):
(2) return 0
(3) return lst[0] + sublist
(4) if not lst:
(5) else:
(6) sublist = sum_list(lst[1:])

If we need to put the above six pieces of code in the correct order to create this function, what is the correct order? (Ignore indentation)

a. (1) (2) (3) (4) (5) (6)
b. (1) (6) (4) (3) (5) (2)
c. (1) (4) (3) (5) (6) (2)
d. (1) (4) (2) (5) (6) (3)
e. None of the above

Grading Rubric

Each multiple-choice question is 1 point.

Grade Point	Letter Grade	Points on Exam
4	A+/A	44 and above
3.7	A-	41 and above
3.3	B+	38 and above
3	B	35 and above
2.7	B-	32 and above
2.3	C+	29 and above
2	C	26 and above
1.7	C-	23 and above
1.3	D+	20 and above
1	D	17 and above
0	F	0 and above