Midterm Examination *1st Semester A.Y. 2022-2023*			**Questionnaire No.:**
College: CICT	**Department:** BSIT	**Course Code:** IT 403 WMAD	**Course Title:** Elective 5
Faculty: Aaron Paul M. Dela Rosa, MSIT, GK-CDPO, PCEP, JSE, CPE, CLE, ITS, MIE, GCE, DEL			
IMPORTANT! 1. Write **questionnaire no.** to your answer sheet. 2. **Do not write anything** on this questionnaire.			

Direction: Shade the circle of your answer/s completely IN YOUR ANSWER SHEET.

I. SITUATIONAL AND FACTUAL PROBLEMS

1. What will happen when you attempt to run the following code?

```
print(Hello, World!)
```

A. The code will print Hellow, World! To the console.
B. The code will raise the TypeError exception.
C. The code will raise the ValueError exception.
D. The code will raise the SyntaxError exception.

2. The top-most Python exception is called:

A. BaseException
B. TopException
C. Exception
D. PythonException

3. Entering the try: block implies that:

A. The block will be omitted
B. All of the instructions from this block will be executed
C. None of the instructions from this block will be executed
D. Some of the instructions from this block may not be executed

4. The meaning of a positional argument is determined by:

A. Its position within the argument list
B. Its connection with existing variables
C. The argument's name specified along with its value
D. Its value

5. Which of the following sentences are true about the code? (Select two answers)

```
nums = [1, 2, 3]
vals = nums
```

A. nums has the same length as vals
B. vals is longer than nums
C. nums and vals are different lists
D. nums and vals are different names of the same list

6. The unnamed except: block:

A. Cannot be used if any named block has been used
B. Must be the last one
C. Can be placed anywhere
D. Must be the first one

7. ASCII is:

A. A standard module name
B. A character name
C. A standard code points
D. A pre-defined variable name

8. Which of the following are examples of Python built-in concrete exceptions? (Select two answers)

A. IndexError
B. ImportError
C. ArithmeticError
D. LookupError

9. The following statement:

```
assert var == 0
```

A. Will stop the program when var == 0
B. Has no effect
C. Will stop the program when var != 0
D. Is erroneous

10. Which of the following variable names are illegal and will cause the SyntaxError exception? (Select two answers)

A. for
B. print
C. In
D. in

11. The following statement:

```
assert 0
```

A. Will stop the program
B. Has no effect
C. Will not stop the program
D. Is erroneous

12. What is the expected behavior of the following program?

```
foo = (1, 2, 3)
foo.index(0)
```

A. It will output 1 to the screen.
B. It will cause a ValueError exception.
C. It will cause a TypeError exception.
D. It will cause a SyntaxError exception.

13. Which of the following snippets shows the correct way of handling multiple exceptions in a single except clause?

A. except: (TypeError, ValueError)
    # Some code
B. except TypeError, ValueError:
    # Some code
C. except (TypeError, ValueError):
    # Some code
D. except: TypeError, ValueError
    # Some code

14. Which of the following lines correctly invokes the function defined below? (Select two answers)

```
def fun(a, b, c=0):
    pass
```

A. fun(b=0, a=0)
B. fun(b=1)
C. fun()
D. fun(0, 1, 2)

15. Take a look at the snippet and choose the true statement:

```
nums = [1, 2, 3]
vals = nums
del vals[:]
```

A. nums has the same length as vals
B. vals is longer than nums
C. nums is longer than vals
D. the snippet will cause a runtime error

II. ANALOGY

16. PROGRAMMING : COMPILER :: SCRIPTING : ______

A. DEBUGGER
B. INTERPRETER
C. SCRIPTER
D. COMPILER

17. * : REPLICATION :: + : ______

A. ADDITION
B. BINDING
C. CONCATENATION
D. INTERPOLATION

18. X = 0 : ASSIGNING :: X, Y = 0, 1 : ______

A. CASTING
B. UNPACKING
C. PACKING
D. UNCASTING

19. FOR : SEQUENCE :: WHILE : ______

A. ITERATION
B. SELECTION
C. CONDITION
D. FUNCTION

20. BRACKETS : LISTS :: BRACES : ______

A. SETS
B. DICTIONARIES
C. TUPLES
D. RANGE

21. IF : TERNARY :: FUNCTION : ______

A. BUILT-IN
B. PRE-DEFINED
C. USER-DEFINED
D. LAMBDA

22. ZERODIVISIONERROR : ARITHMETICERROR :: KEYERROR : ______

A. INDEXERROR
B. TYPEERROR
C. LOOKUPERROR
D. VALUEERROR

23. MAP : TRANSFORM :: FILTER : ______

A. EXTRACT
B. INCLUDE
C. PROCESS
D. PRODUCE

24. DEFINE : PARAMETERS :: INVOKE : ______

A. PARAMETERS
B. IMPLEMENTATION
C. STATEMENTS
D. ARGUMENTS

25. INPUT() : STRING :: PRINT() : ______

A. NONE
B. STRING
C. INTEGER
D. VOID

26. STR : TEXT :: LIST : ______

A. SEQUENCE
B. RANGE
C. COLLECTION
D. MAPPING

27. SEP : ‘ ‘ :: END : ______

A. ‘\NL’
B. ‘\LN’
C. ‘\N’
D. ‘\L’

28. [ ] : INDEXING :: [:] : _____

A. SPLITTING
B. JOINING
C. TRIMMING
D. SLICING

29. TYPE(0.0) : <CLASS ‘FLOAT’> :: TYPE(‘’) : _____

A. <CLASS ‘NONE’>
B. <CLASS ‘INT’>
C. <CLASS ‘STR’>
D. <CLASS ‘STRING’>

30. GENERIC : ABSTRACT :: SPECIFIC : _____

A. OBJECTIVE
B. ERROR
C. EXCEPTION
D. CONCRETE

III. OUTPUT TRACING

31. The following code:

```
print('Aron' > 'Aaron')
```

prints:

A. 0
B. False
C. 1
D. True

32. The following code:

```
print(3 * 'a' + 'p')
```

prints:

A. appp
B. aaap
C. apapap
D. aaappp

33. What is the output of the following piece of code?

```
tup = 1, 2, 4, 8
tup = tup[-2:-1]
tup = tup[-1]
print(tup)
```

A. (4)
B. (4,)
C. 4
D. the snippet will cause a runtime error

34. The following code:

```
print(chr(ord('z')) - 5)
```

prints:

A. v
B. u
C. The snippet will cause a TypeError
D. The snippet will cause a ValueError

35. The following code:

```
print(float('1, 3'))
```

A. Raises a ValueError exception
B. Raises a SyntaxError Exception
C. Raises a TypeError exception
D. Raises an ArithmeticError exception

36. The following code:

```
x = '\''
print(len(x))
```

prints:

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

37. The following code:

```
print(sep='_', 'hello')
```

A. Raises a ValueError exception
B. Raises a SyntaxError Exception
C. Raises a TypeError exception
D. Raises an ArithmeticError exception

38. What is the output of the following piece of code?

```
x = 1 // 5 + 1 / 5
print(x)
```

A. 0.0
B. 0.4
C. 0
D. 0.2

39. The following code:

```
print(ord('p') - ord('l'))
```

prints:

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

40. The following snippet:

```
hi()
def hi():
    print('hi!')
```

A. Will output 'hi!'
B. Will cause a NameError
C. Will cause a TypeError
D. Will output hi!

41-43. What is the output of the following snippet?

```
def fun(x, y):
    if x == y:
        return x
    else:
        return fun(x, y-1)
print(fun(0, 3))
```

A. 2
B. The snippet will cause a runtime error
C. 0
D. 1

44-46. How many stars (*) will the following snippet send to the console?

```
i = 0
while i < i + 2:
    i += 1
    print("*")
else:
    print("*")
```

A. It's an infinite loop, printing one star per line
B. Zero
C. One
D. Two

47-49. How many hashes (#) will the following snippet send to the console?

```
lst = [[x for x in range(3)] for y in range(3)]
for r in range(3):
    for c in range(3):
        if lst[r][c] % 2 != 0:
            print("#")
```

A. zero
B. three
C. six
D. nine

50-52. What is the output of the following snippet?

```
dct = {'one': 'two', 'three': 'one', 'two': 'three'}
v = dct['three']
for k in range(len(dct)):
    v = dct[v]
print(v)
```

A. ('one', 'two', 'three')
B. two
C. three
D. one

53-55. The following snippet:

```
def func(a, b):
    return b ** a
print(func(b=2, 2))
```

A. Will output None
B. Will output 4
C. Will output 2
D. Is erroneous

56-58. What is the output of the following code if the user enters a 0?

```
try:
    v = input()
    print(int(v)/len(v))
except ValueError:
    print("Bad input")
except:
    print("Booo!")
```

A. 0.0
B. Bad input
C. Booo!
D. 1.0

59-61. What is the output of the following code?

```
dct = {}
dct['1'] = (1, 2)
dct['2'] = (2, 1)
for x in dct.keys():
    print(dct[x][1], end="")
```

A. 12
B. (1, 2)
C. (2, 1)
D. 21

62-64. What is the output of the following code?

```
x = 1
y = 2
x, y, z = x, x, y
z, y, z = x, y, z
print(x, y, z)
```

A. 1 2 1
B. 1 1 2
C. 2 1 2
D. 1 2 2

65-67. What is the output of the following code?

```
try:
    print("5"/0)
except ArithmeticError:
    print("arith")
except ZeroDivisionError:
    print("zero")
except:
    print("some")
```

A. zero
B. 0
C. arith
D. some

68-70. The following snippet:

```
def fun1(a):
    return None
def fun2(a):
    return fun1(a) * fun1(a)
print(fun2(2))
```

A. Will output 4
B. Will cause a ValueError
C. Will cause a TypeError
D. Will output 16

71-75. What is the output of the following code?

```
def fun(x):
    return 1 if x % 2 == 0 else

print(fun(fun(2)) + 1)
```

A. None
B. 3
C. TypeError
D. SyntaxError

76-80. What is the output of the following code?

```
def f(x):
    try:
        x = x / x
    except:
        print("a", end='')
    else:
        print("b", end='')
    finally:
        print("c", end='')

f(1)
f(0)
```

A. It will print bcbc
B. It will print acac
C. It will print bcac
D. It will raise an unhandled exception

81-85. What is the output of the following code?

```
class A:
    def __init__(self, v = 1) -> None:
        self.v = v

    def set(self, v) -> int:
        self.v = v
        return v

a = A()
print(a.set(a.v + 1))
```

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

86-90. What is the output of the following code?

```
class A:
    def __str__(self) -> str:
        return 'a'

class B(A):
    def __str__(self) -> str:
        return 'b'

class C(B):
    pass

o = C()
print(o)
```

A. It will print a

B. It will raise an exception

C. It will print c

D. It will print b

91-95. What is the output of the following code?

```
class A:
    def __init__(self, v) -> None:
        self.__a = v + 1

a = A(0)
print(a.__a)
```

A. It will print 0

B. It will print 1

C. It will raise a NameError exception

D. It will raise an AttributeError exception

96-100. What is the output of the following code?

```
class A:
    v = 2

class B(A):
    v = 1

class C(B):
    pass

o = C()
print(o.v)
```

A. It will print 2

B. It will print 1

C. It will raise an exception

D. It will print an empty line

IV. FILL IN THE MISSING CODE

101-105. Which of the following missing lines of code should be used to have the desired output?

```
PYTHONINTHEWEB
```

Code:

```
str = 'aQsddeZghUtuImuPfOvJgyOkUrvIbFfgyhXgFrfdvvCg'
for s in str:
    if s.isupper():
        # Missing code here
```

A. `print(chr(ord(s)+1),end="")`

B. `print(chr(ord(s)-1),end="")`

C. `print(ord(chr(s)+1),end="")`

D. `print(ord(chr(s)-1),end="")`

106-110. Which of the following missing lines of code should be used to have the desired output?

```
15
```

Code:

```
def f(x):
    if x == 0:
        # Missing code here
    return x + f(x - 1)
print(f(5))
```

A. `return 0`

B. `return 1`

C. `return None`

D. `return True`

111-115. Which of the following missing lines of code should be used to have the desired output?

```
['aaron']: a: 2 a: 2 r: 1 o: 1 n: 1
['paul']: p: 1 a: 1 u: 1 l: 1
```

Code:

```
for w in "aaron paul".split():
      # Missing code here
      for l in list(w):
            print(f" {l}: {w.count(l)}", end="")
      print()
```

A. `print(f"[{w}]:")`

B. `print(f"{[w]}:", end="")`

C. `print(f"[{w}]:", end="")`

D. `print(f"{[w]}:")`

116-120. Which of the following missing lines of code should be used to have the desired output?

```
a
b
c
```

Code:

```
dct = {}
lst = ['a', 'b', 'c', 'd']
for i in range(len(lst)-1):
    dct[lst[i]] = lst[i],
for i in sorted(dct.keys()):
    k = dct[i]
    # Missing code here
```

A. `print(k)`
B. `print(k[i])`
C. `print(k["0"])`
D. `print(k[0])`

121-125. Which of the following missing lines of code should be used to have the desired output?

```
['a', 'a', 'ror', 'o', 'non', ' ', 'pop', 'a', 'u', 'lol']
```

Code:

```
def alienLanguage(msg):
    # Missing code here
print(alienLanguage("Aaron Paul"))
```

A. `return list(map(lambda a: a+"o"+a if a not in 'aeiou ' else a, msg.lower()))`
B. `return list(map(lambda a: a+o+a if a not in 'aeiou ' else a, msg.lower()))`
C. `return list(map(lambda a: a+"o"+a if a not in 'aeiou' else a, msg.lower()))`
D. `return list(map(lambda a: a+o+a if a not in 'aeiou' else a, msg.lower()))`