| Cairo University | Cairo University<br>Faculty of Graduate Studies for Statistical Researches<br>**Department:** Computer Sciences<br>**Academic Year:** 2021-2022 **Semester:** Second<br>**Date:** 12/ 6/2022 **Level:** Diploma | |
|---|---|---|

| **Course Title:** | **Course code:** CS522 | **Time:** 2 Hours | **Exam marks:** 100 | **# Exam. Sheets:** 13 Pages |
|---|---|---|---|---|
| **Exam. Instructions : ANSWER THE FOLLOWING QUESTIONS in Python** | | | | نموذج: ( أ ) |

## Question 1:

**Choose the correct answer for each of the following:**

1. Which of the following **assign** an **integer** value to a **variable** correctly?

   ***(a)*** x=25 ***(b)*** x='hello' ***(c)*** 2x=5 ***(d)*** 2x='hello'

---

2. Which of the following is a **loop without body**?

***(a)***
```
x=3
while(x>3)
```

***(b)***
```
x=3
while(x>3);
```

***(c)***
```
x=3
while(x>3)
{}
```

***(d)***
```
x=3
while(x>3):
    pass
```

---

3. Which of the following is a correct **comment** in python?

***(a)*** #comment ***(b)*** '''comment

***(c)*** ?comment ***(d)*** */comment/*

---

4. What is the index number of the **last element** of a **tuple** with **50 elements**?

***(a)*** 50 ***(b)*** 49 ***(c)*** -1 ***(d)*** b and c

---

5. The **last printed expression** is assigned to the ..... variable

***(a)*** last ***(b)*** _ ***(c)*** var ***(d)*** exp

---

6. What is the **output** of the following:

```
A=25
print(a)
```

***(a)*** error ***(b)*** 25 ***(c)*** a ***(d)*** A

---

7. Which of the following is used to **initialize multiple variables** with a **common value**?

***(a)*** x = y: y = 33 ***(b)*** x = y = z = 33

***(c)*** x = z; y = z; x = 33; ***(d)*** x & y & z = 33

---

8.
```
a = input("enter your input: ")
print(type(a) )
```

***(a)*** int ***(b)*** string ***(b)*** float ***(d)*** bool

---

9.
```
x = '24' + '16'
print(x)
```

***(a)*** 40 ***(b)*** x ***(b)*** 21 ***(d)*** 2416

---

10. ..... can **store different types** of values

***(a)*** variable ***(b)*** list ***(c)*** function ***(d)*** a and c

---

## Question 2:

**Choose the equivalent code for each of the following:**

11.
```
x=3
if x>3:
   print('x>3')
elif x<3:
   print('x < 3')
else:
   print('x = 3')
```

***(a)***
```
x=3
if x>3 print('x>3') elif x<3 print('x<3') else print('x = 3')
```

***(b)***
```
x=3
if x>3 print('x>3') else if x<3 print('x<3') print('x = 3') else
```

***(c)***
```
x=3
print('x>3') if x>3    else print('x<3') if x<3    else print('x = 3')
```

***(d)***
```
x=3
print('x>3') if x> 3    if x<3 print ('x<3')      print('x = 3')    else
```

---

12.
```
d={'k1': 25, 'k2': 46}
for k, v in d.items():
      print(k, v)
```

***(a)***
```
d={'k1': 25, 'k2': 46}
for k in d:
      print(k, d[k])
```

***(b)***
```
d={'k1': 25, 'k2': 46}
for k in d.items():
      print(d)
```

***(c)***
```
d={'k1': 25, 'k2': 46}
for k in d:
      print(k)
```

***(d)***
```
d={'k1': 25, 'k2': 46}
for k in d:
      print(d[k])
```

---

13.

```
num=35
if num%2==0:
    print('even')
else:
    print('odd')
```

***(a)***

```
num=35
print([num%2==0]('odd', 'even'))
```

***(b)***

```
num=35
print([num%2==0]('even', 'odd'))
```

***(c)***

```
num=35
print(('even', 'odd')[num%2==0])
```

***(d)***

```
num=35
print(('odd', 'even')[num%2==0])
```

---

14.

```
alpha=('a', 'b', 'c', 'd')
for i in range(len(alpha)):
    print(alpha[i])
```

***(a)***

```
alpha= ('a', 'b', 'c', 'd')
for i in range(len(alpha)):
    print(i)
```

***(b)***

```
alpha= ('a', 'b', 'c', 'd')
for i in range(len(alpha)):
    print(alpha)
```

***(c)***

```
alpha= ('a', 'b', 'c', 'd')
for i in alpha:
    print(i)
```

***(d)***

```
alpha= ('a', 'b', 'c', 'd')
for i in range(len(alpha)):
    pass
```

---

15.

```
x=3
z=[]
for i in range(3):
        z[i:]=[i]
```

*(a)*
```
x=3
z =[ i    for i in range(3)    ]
```

*(b)*
```
x=3
z =[ i ]
```

*(c)*
```
x=3
z = for i in range(3) [ i ]
```

*(d)*
```
x=3
z =[ for i in range(3)    i ]
```

---

16.

```
z=[ ]
z[0:]=3
```

*(a)*
```
z=[ ]
z[0]=3
```

*(b)*
```
z=[ ]
z+=(3)
```

*(c)*
```
z=[ ]
z.append(3)
```

*(d)*
```
z=[ ]
z[-1]=3
```

## Question 3:

**Choose the error line number in each of the following:**

17.

```
1. x=12
2. y=4
3. fun(x, y)

4. def fun(x, y):
5.         print(x, y)
```

*(a)* line 1 *(b)* line 2 *(c)* line 5 *(d)* line 3

---

18.

```
1. def calculate(a, b, c=[], d):
2.         return a+b-d

3. w=x=3
4. y, z=5, [1, 2]
5. print(calculate(x, y, z, w)
```

*(a)* line 1 *(b)* line 2 *(c)* line 3 *(d)* line 4

---

19.

```
1. def square(**nums, x):
2.       sqnums={}
3.       for num, val in nums.items():
4.             sqnums[num] = val**2
5.       return sqnums

6. x=square(q=2, f=4)
7. print(x)
```

*(a)* line 1 *(b)* line 2 *(c)* line 3 *(d)* line 4

---

20.

```
1. def foo(a, b):
2.         c=a+b
3.         return c
4. x=3
5. print(foo(x))
```

***(a)*** line 2 ***(b)*** line 3 ***(c)*** line 4 ***(d)*** line 5

---

21.

```
1. def fun (*arg):
2.         x=len(arg)
3.         for i in range(x):
4.                 print(arg[i])
5. fun(2, 3, 4, a=5)
```

***(a)*** line 2 ***(b)*** line 3 ***(c)*** line 4 ***(d)*** line 5

---

22.

```
1. s={1, 2, 3}
2. s.add(3)
3. print(s[0])
4. print(9 in s)
```

***(a)*** line 1 ***(b)*** line 2 ***(c)*** line 3 ***(d)*** line 4

---

23.

```
1. d={'a': 123, 'b': 980}
2. d[3]=23
3. print(5 in d)
4. print(d[0])
```

***(a)*** line 1 ***(b)*** line 2 ***(c)*** line 3 ***(d)*** line 4

---

24.

```
1. def exp (*nums, x):
2.     for i in nums:
3.         print(i)

4. exp(1, 4, 5, 7)
```

***(a)*** line 1 ***(b)*** line 2 ***(c)*** line 3 ***(d)*** line 4

---

25.

```
1. x=5
2. if x>0:
3. print('positive')
4. else:
5. print('negative')
```

***(a)*** line 1 ***(b)*** line 2 ***(c)*** line 3 ***(d)*** line 4

---

26.

```
1. x={}
2. x.add(2)
3. y=x
4. print(x)
```

***(a)*** line 1 ***(b)*** line 2 ***(c)*** line 3 ***(d)*** line 4

## Question 4:

**Choose the the correct output each of the following:**

27.

```
def myfun (a, b, c=[1, 2, 3]):
    c.append(a)
    c.append(b)
    return c
x, y=4, 8
print(myfun(b=x, a=y))
```

***(a)*** [1, 2, 3, 8, 4] ***(b)*** no output ***(c)*** [1, 2, 3] ***(d)*** error

---

28.

```
if -50 in range(-1, -100):
    print('Yes')
else:
    print('No')
```

***(a)*** Yes ***(b)*** No ***(c)*** error ***(d)*** output

---

29.

```
def foo(*grades):
    print(len(grades))
    print(type(grades))
foo(96, 82, 70)
```

***(a)*** 3 tuple ***(b)*** 6 ***(c)*** 3 Dictionary ***(d)*** 6 tuple

---

30.
```
x=lambda a: a**2
print(x(3))
```

***(a)*** 6 ***(b)*** 3 ***(c)*** 9 ***(d)*** 5

---

31.
```
t1=(9, 5, 0)
t2='w', 'n', 'f'
print(t1+t2)
```

***(a)*** 9 5 0 w n f ***(b)*** t1+t2 ***(c)*** 9 5 0 ***(d)*** w n f

---

32.
```
x=5
y=8
txt= “the value of {2} + {1} is {0}”. format(x+y, y, x)
print(txt)
```

***(a)*** the value of {2} + {1} is {0} ***(b)*** the value of x + y is x+y

***(c)*** the value of 5 + 8 is 13 ***(d)*** the value of 13 + 8 is 5

---

33.
```
x=[0, 1, 2, 3, 4, 5, 6, 7]

print(x[1 : 6 : 2])
```

***(a)*** 1 3 5 ***(b)*** 0 1 2 3 4 ***(c)*** 1 2 3 4 ***(d)*** 0 2 4

---

34.
```
z={6, 9, 3}
z.remove ('b')
print(z)
```

***(a)*** error ***(b)*** no output ***(c)*** {9, 3, 6} ***(d)*** {9, 3, 6, ‘b’}

---

35.

```
def square(sqr, nums):
    for n, val in nums.items():
        sqr[n] = val**2

h={}
x={'a': 2, 'b':1}
square(h, x)
print(h)
```

***(a)*** **{ }** ***(b)*** {'a': 4, 'b': 1} ***(c)*** no output ***(d)*** h

---

36.

```
h = [3, 2, 1]
m = h[:]
if id(h) == id(m): print("yes")
else: print("No")
```

***(a)*** yes ***(b)*** No ***(c)*** Error ***(d)*** no output

---

37.

```
def foo(x):
    for i in range (2):
        x.append(i)
x=[]
foo(x)
print(x)
```

***(a)*** [ ] ***(b)*** [0, 1] ***(c)*** x ***(d)*** no output

---

38.

```
def sum (n):
    if n==1: return 1
    return n + sum(n-1)
print(sum(3))
```

***(a)*** 4 ***(b)*** 10 ***(c)*** 6 ***(d)*** [5 3 9]

---

39.

```
x=5
while(x<9):
    x+=1
    if x>5: break
else: print(x)
```

***(a)*** error ***(b)*** no output ***(c)*** 6 ***(d)*** 8

---

40.

```
def fun(s):
      s.append('world')

e=['hello']
fun(e)
print(e)
```

***(a)*** e ***(b)*** ['hello'] ***(c)*** ['hello', 'world'] ***(d)*** ['world']

**remove()** remove an item from a set. If the item to remove does not exist, remove() will raise an error.

**set**() creates a set.

**add**() adds an item to a set.

**items()** returns a list of dictionary's (key, value) tuple pairs.

**len(x)** returns the number of items in the collection x.

**range(n)** generates a sequence of numbers from zero to n-1.

**append(n)** add a single element to the end of the list

**format()** takes the passed arguments, formats them, and places them in the string where the placeholders {} are