1. Select all the correct options to join two lists in Python

listOne = ['a', 'b', 'c', 'd']

listTwo = ['e', 'f', 'g']

 newList = listOne + listTwo

 newList = extend(listOne, listTwo)

 newList = listOne.extend(listTwo)

 newList.extend(listOne, listTwo)

2. What is the output of the following code

aList = ["PYnative", [4, 8, 12, 16]]

**print**(aList[0][1])

**print**(aList[1][3])

 P 8  
Y 16

 P  
12

 Y  
16

3. What is the output of the following list operation

sampleList = [10, 20, 30, 40, 50]

**print**(sampleList[-2])

**print**(sampleList[-4:-1])

 40  
[20, 30, 40]

 IndexError: list index out of range

4. What is the output of the following list operation

aList = [10, 20, 30, 40, 50, 60, 70, 80]

**print**(aList[2:5])

**print**(aList[:4])

**print**(aList[3:])

 [20, 30, 40, 50]  
[10, 20, 30, 40]  
[30, 40, 50, 60, 70, 80]

 [30, 40, 50]  
[10, 20, 30, 40]  
[40, 50, 60, 70, 80]

5. What is the output of the following list function?

sampleList = [10, 20, 30, 40, 50]

sampleList.append(60)

**print**(sampleList)

sampleList.append(60)

**print**(sampleList)

 [10, 20, 30, 40, 50, 60]  
[10, 20, 30, 40, 50, 60]

 [10, 20, 30, 40, 50, 60]  
[10, 20, 30, 40, 50, 60, 60]

6. What is the output of the following code

my\_list = ["Hello", "Python"]

**print**("-".join(my\_list))

 HelloPython-

 Hello-Python

 -HelloPython

7. In Python, list is mutable

 False

 True

8. What is the output of the following list comprehension

resList = [x+y **for** x **in** ['Hello ', 'Good '] **for** y **in** ['Dear', 'Bye']]

**print**(resList)

 [‘Hello Dear’, ‘Hello Bye’, ‘Good Dear’, ‘Good Bye’]

 [‘Hello Dear’, ‘Good Dear’, ‘Hello Bye’, ‘Good Bye’]

9. What is the output of the following

aList = [5, 10, 15, 25]

**print**(aList[::-2])

 [15, 10, 5]

 [10, 5]

 [25, 10]

10. What is the output of the following code

list1 = ['xyz', 'zara', 'PYnative']

**print** (**max**(list1))

 PYnative

 zara

11. What is the output of the following

aList = [1, 2, 3, 4, 5, 6, 7]

pow2 = [2 \* x **for** x **in** aList]

**print**(pow2)

 [2, 4, 6, 8, 10, 12, 14]

 [2, 4, 8, 16, 32, 64, 128]

12. What is the output of the following code?

sampleList = [10, 20, 30, 40]

**del** sampleList[0:6]

**print**(sampleList)

 []

 list index out of range.

 [10, 20]

13. What is the output of the following list assignment

aList = [4, 8, 12, 16]

aList[1:4] = [20, 24, 28]

**print**(aList)

 [4, 20, 24, 28, 8, 12, 16]

 [4, 20, 24, 28]

14. What is the output of the following

l = [None] \* 10

**print**(**len**(l))

 10

 0

 Syntax Error

15. Select all the correct options to copy a list

aList = ['a', 'b', 'c', 'd']

 newList = copy(aList)

 newList = aList.copy()

 newList.copy(aList)

 newList = list(aList)

16. What is the output of the following list function?

sampleList = [10, 20, 30, 40, 50]

sampleList.pop()

**print**(sampleList)

sampleList.pop(2)

**print**(sampleList)

 [20, 30, 40, 50]  
[10, 20, 40]

 [10, 20, 30, 40]  
[10, 20, 30, 50]

 [10, 20, 30, 40]  
[10, 20, 40]

17. What is the output of the following set operation

set1 = {"Yellow", "Orange", "Black"}

set2 = {"Orange", "Blue", "Pink"}

set3 = set2.difference(set1)

**print**(set3)

 {‘Yellow’, ”Black’, ‘Pink’, ‘Blue’}

 {‘Pink’, ‘Blue’}

 {‘Yellow’, ”Black’}

18. What is the output of the following

set1 = {10, 20, 30, 40, 50}

set2 = {60, 70, 10, 30, 40, 80, 20, 50}

**print**(set1.issubset(set2))

**print**(set2.issuperset(set1))

 False  
False

 True  
True

19. The union() method returns a new set with all items from both sets by removing duplicates

 True

 False

20. What is the output of the following

sampleSet = {"Yellow", "Orange", "Black"}

sampleSet.discard("Blue")

**print**(sampleSet)

 {‘Yellow’, ‘Orange’, ‘Black’}

 KeyError: ‘Blue’

21. What is the output of the following code

sampleSet = {"Yellow", "Orange", "Black"}

**print**(sampleSet[1])

 Yellow

 Syntax Error

 Orange

22. What is the output of the following set operation

sampleSet = {"Yellow", "Orange", "Black"}

sampleSet.update(["Blue", "Green", "Red"])

**print**(sampleSet)

 {‘Yellow’, ‘Orange’, ‘Red’, ‘Black’, ‘Green’, ‘Blue’}

 {‘Yellow’, ‘Orange’, ‘Black’, [“Blue”, “Green”, “Red”]}

 TypeError: update() doesn’t allow list as a argument.

23. The symmetric\_difference() method returns a set that contains all items from both sets, but not the items that are present in both sets.

 False

 True

24. Select all the correct ways to copy two sets



* set2 = set1.copy()
* set2 = set(set1)
* set2.update(set1)



* set2 = set1

25. What is the output of the following code

aSet = {1, 'PYnative', ['abc', 'xyz'], True}

**print**(aSet)

 {1, ‘PYnative’, [‘abc’, ‘xyz’]}

 {1, ‘PYnative’, [‘abc’, ‘xyz’], True}

 TypeError

26. Select all which is true for Python set



* Sets are unordered
* set doesn’t allow duplicate
* sets are written with curly brackets {}



* set object does support indexing
* set is mutable

27. What is the output of the following union operation

set1 = {10, 20, 30, 40}

set2 = {50, 20, "10", 60}

set3 = set1.union(set2)

**print**(set3)

 {40, 10, 50, 20, 60, 30}

 {40, ’10’, 50, 20, 60, 30}

 {40, 10, ’10’, 50, 20, 60, 30}

 SynatxError: Different types cannot be used with sets

28. The isdisjoint() method returns True if none of the items are present in both sets, otherwise, it returns False.

 True

 False

29. What is the output of the following code

aSet = {1, 'PYnative', ('abc', 'xyz'), True}

**print**(aSet)

 TypeError

 {‘PYnative’, 1, (‘abc’, ‘xyz’), True}

 {‘PYnative’, 1, (‘abc’, ‘xyz’)}

30. What is the output of the following set operation.

set1 = {"Yellow", "Orange", "Black"}

set2 = {"Orange", "Blue", "Pink"}

set1.difference\_update(set2)

**print**(set1)

 {‘Black’, ‘Yellow’}

 {‘Yellow’, ‘Orange’, ‘Black’, ‘Blue’, ‘Pink’}

31. What is the output of the following

sampleSet = {"Yellow", "Orange", "Black"}

sampleSet.add("Blue")

sampleSet.add("Orange")

**print**(sampleSet)

 {‘Blue’, ‘Orange’, ‘Yellow’, ‘Orange’, ‘Black’}

 {‘Blue’, ‘Orange’, ‘Yellow’, ‘Black’}

32. Select all the correct options to remove “Orange” from the set.

sampleSet = {"Yellow", "Orange", "Black"}

 sampleSet.pop("Orange")

 sampleSet.discard("Orange")

 del sampleSet ["Orange"]