Multiple Choice

1. You can use the operator to determine whether a key exists in a dictionary.

a. & b. in c. ˆ d. ?

2. You use to delete an element from a dictionary.

a. the remove method b. the erase method c. the delete method d. the del statement

3. The function returns the number of elements in a dictionary.

a. size() b. len() c. elements() d. count()

4. You can use to create an empty dictionary.

a. {} b. () c. [] d. empty()

5. The method returns a randomly selected key-value pair from a dictionary.

a. pop() b. random() c. popitem() d. rand_pop()

6. The method returns the value associated with a specified key and removes

that key-value pair from the dictionary. a. pop() b. random() c. popitem() d. rand_pop()

7. The dictionary method returns the value associated with a specified key.

If the key is not found, it returns a default value. a. pop() b. key() c. value() d. get()

8. The method returns all of a dictionary's keys and their associated values as

a sequence of tuples. a. keys_values() b. values() c. items() d. get()

9. The following function returns the number of elements in a set:

a. size() b. len() c. elements() d. count()

10. You can add one element to a set with this method.

a. append b. add c. update d. merge

11. You can add a group of elements to a set with this method.

a. append b. add c. update d. merge

12. This set method removes an element, but does not raise an exception if the element is

not found. a. remove b. discard c. delete d. erase

13. This set method removes an element and raises an exception if the element is not

found. a. remove b. discard c. delete d. erase

14. This operator can be used to find the union of two sets.

a. | b. & c. − d. ˆ

15. This operator can be used to find the difference of two sets.

a. | b. & c. − d. ˆ

16. This operator can be used to find the intersection of two sets.

a. | b. & c. − d. ˆ

17. This operator can be used to find the symmetric difference of two sets.

a. | b. & c. – d. ˆ

True or False

1. The keys in a dictionary must be mutable objects.

False

2. Dictionaries are not sequences.

True

3. A tuple can be a dictionary key.

True

4. A list can be a dictionary key.

False

5. The dictionary method popitem does not raise an exception if it is called on an empty dictionary. False

6. The following statement creates an empty dictionary:

mydct = {} True

7. The following statement creates an empty set:

myset = () False

8. Sets store their elements in an unordered fashion.

True

9. You can store duplicate elements in a set.

False

10. The remove method raises an exception if the specified element is not found in the set True

Short Answer

1. 2
2. 1
3. Not found
4. It’ll raise a KeyError because 2 isn’t a key in the months dictionary.
5. You can use the del statement, like this: del dictionary[key]
6. Use the len() function on the dictionary
7. 4
8. It’ll print: 1 2 3 (Maybe in a different order though, since dictionaries aren’t ordered)
9. {‘S’, ‘a’, ‘t’, ‘u’, ‘r’, ‘n’}
10. This will cause an error. You can’t make a set from an integer.
11. {‘a’, ‘ ‘, ‘b’, ‘c’, ‘d’}

12. {2, 4, 6}
13. {‘a’, ‘bb’, ‘ccc’, ‘dddd’}
14. 5
15. {10, 20, 30, 40, 50, 60}
16. {‘p’, ‘s’} These are the elements that appear in both set1 and set2.
17. {‘f’} This is the element in set1 that’s not in set2.
18. {‘a’, ‘b’, ‘c’} These are the elements in set2 that aren’t in set1.
19. {1, 4} These are the elements that are in either set1 or set2, but not in both.
20. The intersection of set1 and set2 would be an empty set {}, because they have no common elements.

    The union of set1 and set2 would give the same result as their symmetric difference in this case, because they have no elements in common. Both operations would result in {‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’}.