Python MCQS Day3

Subhan Kaladi

1. **What is a key characteristic of a Python dictionary?**
   A) It is ordered and immutable
   B) It stores data in key-value pairs and is mutable
   C) It allows duplicate keys
   D) It is a collection of ordered items only
2. **Given the dictionary `dict = {"name": "Kaladi", "cgpa": 9.0, "mark": 85}` what will dict["name"] return?**
   A) 9.0
   B) 85
   C) Kaladi
   D) KeyError
3. **What happens if you try to add a duplicate key to a dictionary, as in `dict = {"name": "Subhan", "name": "Kaladi"}`?**
   A) It raises a KeyError
   B) It keeps both values
   C) It updates the value to the latest one ("Kaladi")
   D) It ignores the duplicate key
4. **Consider the nested dictionary `student = {"name": "Kaladi", "score": {"chem": 98, "phy": 97, "math": 95}}`. How do you access the value 95?**
   A) student["score"]["math"]
   B) student["math"]
   C) student["score"][2]
   D) student[1]["math"]
5. **What does the dictionary method `myDict.keys()` return?**
   A) All values in the dictionary
   B) All keys in the dictionary
   C) A single key-value pair
   D) The last key added

6. **What will `myDict.get("key")` return if "key" does not exist in myDict?**

A) KeyError
B) None
C) 0
D) Empty string

7. **What is the output of the following code?**

```
dict = {"a": 1, "b": 2}

dict["c"] = 3

print(dict)
```

A) {"a": 1, "b": 2}
B) {"a": 1, "b": 2, "c": 3}
C) {"c": 3}
D) Error

8. Which of the following is a valid way to create an empty dictionary?

A) dict = {}
B) dict = []
C) dict = ()
D) dict = set()

9. What is a key property of a Python set?

A) It allows duplicate elements
B) It is a collection of ordered items
C) It contains unique and immutable elements
D) It is mutable but ordered

10. Given the set `nums = {1, 2, 3, 4}`, what will `nums.add(5)` do?

A) Raise an error
B) Add 5 to the set
C) Replace an existing element
D) Do nothing

11. What is the output of the following code?

```
set2 = {1, 2, 2, 2}

print(set2)
```

A) {1, 2, 2, 2}
B) {1, 2}
C) {2}
D) Error

12. What does the `set.remove(el)` method do?

A) Adds an element to the set
B) Removes the specified element, raises KeyError if not found
C) Removes a random element
D) Clears the entire set

13. What will `set.clear()` do to a set?

A) Remove one element
B) Empty the set
C) Return a copy of the set
D) Raise an error

14. What is the output of the following code?

```
set1 = {1, 2, 3}

set2 = {2, 3, 4}

print(set1.union(set2))
```

A) {2, 3}
B) {1, 2, 3, 4}
C) {1, 4}
D) Error

15. What does `set1.intersection(set2)` return for `set1 = {1, 2, 3}` and `set2 = {2, 3, 4}`?

A) {1, 2, 3, 4}
B) {2, 3}
C) {1, 4}
D) Empty set

16. How would you store the meanings of "table" and "cat" in a dictionary as per the practice question?

```
dict = {

  "table": "a piece of furniture",

  "cat": "a small animal"
```

}

A) Correct as shown
B) Use lists instead of strings
C) Use tuples as keys
D) Use sets as values

17. Given the list subjects = ["python", "java", "C++", "python", "javascript", "java", "python", "java", "C++", "c"], how many unique subjects (classrooms) are needed?

A) 10
B) 5
C) 6
D) 4

18. What is the correct way to create an empty set in Python?

A) set = {}
B) set = set()
C) set = []
D) set = ()

19. What will the following code output?

```
myDict = {}

myDict["math"] = 90

myDict["phy"] = 85

myDict["chem"] = 88

print(myDict)
```

A) {"math": 90, "phy": 85, "chem": 88}
B) ["math": 90, "phy": 85, "chem": 88]
C) Error
D) {}

20. How can you store 9 and 9.0 as separate values in a set?

A) Use {9, 9.0} directly
B) Use {str(9), 9.0}
C) Use {9, float(9)}
D) Not possible

21. What will `myDict.values()` return for `myDict = {"a": 1, "b": 2, "c": 3}`?

A) ["a", "b", "c"]
B) [1, 2, 3]
C) [("a", 1), ("b", 2), ("c", 3)]
D) Error

22. What happens if you try to access a non-existent key using myDict["key"]?

A) Returns None
B) Raises KeyError
C) Returns 0
D) Creates a new key

23. What is the output of the following code?

```
nums = {1, 2, 3}

nums.pop()

print(len(nums))
```

A) 3
B) 2
C) 1
D) Error

24. Which method can be used to combine two sets and keep only common elements?

A) union()
B) intersection()
C) add()
D) pop()

25. What is the output of the following code?

```
dict = {"name": "Kaladi", "score": {"chem": 98}}

print(dict["score"]["chem"])
```

A) Kaladi
B) 98
C) {"chem": 98}
D) Error

Answer Key

1. B
2. C
3. C
4. A
5. B
6. B
7. B
8. A
9. C
10. B
11. B
12. B
13. B
14. B
15. B
16. A
17. B
18. B
19. A
20. C
21. B
22. B
23. B
24. B
25. B

Thank You