

Quiz 03 - Practice

COMP 110: Introduction to Programming
Spring 2025

Friday, March 21, 2025

Name: _____

9-digit PID: _____

Do not begin until given permission.

Honor Code: I have neither given nor received any unauthorized aid on this quiz.

Signed: _____

Question 1: Multiple Choice For each of the next questions, select all of `set`, `list`, and/or `dict` for which the statement describes. Bubble in ALL squares that apply.

- | | |
|---|---|
| <p>1.1. Which of the following data structures are sequences?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.2. Select all data structures that are mutable.
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.3. Select all data structures that can contain duplicate values.
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.4. Which of these data structures use key-value pairs for storing data?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.5. Which of the following data structures does not guarantee the order of elements? (The <code>dict</code> data structure is intentionally omitted; in Python, order is maintained. However, generally, <code>dict</code>-like data structures do not guarantee ordering.)
☐ <code>list</code> ☐ <code>set</code></p> <p>1.6. Which data structures allow indexing via subscription notation to access individual elements directly?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.7. If you need to store a collection of items and frequently check whether an item is in the collection, which data structure is most efficient?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.8. To ensure the order of elements is maintained and allow for duplicates, which data structure would you choose?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.9. Which of the following data structures require the <code>.add()</code> method to add a value?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.10. To store a sequence of elements that you intend to iterate over and modify, which data structure offers the best performance?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> | <p>1.11. For associating student PIDs to their respective email addresses, which data structure provides the most efficient lookup?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.12. Which data structure's <i>literal syntax</i> is enclosed within curly braces?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.13. Which data structure's <i>literal syntax</i> is enclosed within square brackets?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.14. Which data structures can you iterate over using a <code>for...in</code> loop?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.15. Which data structures allow the use of the <code>len</code> function to determine the <i>number of elements</i> it contains?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.16. Which of the following data structures is best when you want to find the <i>intersection</i>, <i>union</i>, or <i>difference</i> between two collections of values?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.17. If you were creating a messaging app, where you want to maintain a list of messages in the order they were received, which data structure would you use?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.18. When trying to count the frequency of words in a document, which data structure would allow you to efficiently store and update counts?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> <p>1.19. If you want to specify the data type with which a collection of values is indexed, which data structure should you use?
☐ <code>list</code> ☐ <code>set</code> ☐ <code>dict</code></p> |
|---|---|

Question 2: Looping Short Answer Consider the following dictionary and set. For each code sample below, write the corresponding output. Separate lines of output can be separated by a comma. If the code would raise an error, please write "error."

```
1 vend: dict[str, str] = {"A1": "Oreos", "A2": "Lays", "B1": "Coke", "B2": "7up"}
2 flavors: set[str] = {"Orange", "Cherry", "Lime"}
```

2.1. What will be printed?

```
1 for prod in vend:
2     print(prod)
```

2.2. What will be printed?

```
1 for prod in vend:
2     print(vend[prod])
```

2.3. What will be printed?

```
1 for flav in flavors:
2     print(flav)
```

2.4. What will be printed?

```
1 if "Berry" in flavors:
2     print("Available!")
3 else:
4     print("Out...")
```

2.5. What will be printed?

```
1 def buy(vm: dict[str, str]) -> str:
2     for thing in vm:
3         return thing
4     return "Other"
5
6 print(buy(vm=vend))
```

Question 3: Respond to the following questions. Consider the following dictionary:

```
1 vend: dict[str, str] = {"A1": "Oreos", "A2": "Lays", "B1": "Coke", "B2": "7up"}
```

3.1. Write a line of code to find the length of the **vend** dictionary.

3.2. Write a line of code to add the key-value pair, "B3" and "Fanta", to the dictionary.

3.3. Write a line of code to change the value associated with the key, "A1", to "Twix".

3.4. Write a line of code to remove the key-value pair, "A2" and "Lays", from the dictionary.

Question 4: Memory Diagram Trace a memory diagram of the following code listing.

```
1 def count(xs: list[int]) -> dict[int, int]:
2     counts: dict[int, int] = {}
3     for x in xs:
4         if x in counts:
5             counts[x] += 1
6         else:
7             counts[x] = 1
8     return counts
9
10
11 numbers: list[int] = [1, 1, 0]
12 print(count(numbers))
```

Output

Stack

Heap

Globals

Question 5: Memory Diagram Trace a memory diagram of the following code listing.

```
1 def artist_counts(playlist: dict[str, str]) -> dict[str, int]:
2     artists: dict[str, int] = dict()
3     for track in playlist:
4         art: str = playlist[track]
5         if playlist[track] not in artists:
6             artists[art] = 1
7         else:
8             artists[art] += 1
9     return artists
10
11 songs: dict[str, str] = {
12     "B2b": "Charli",
13     "Hello": "Erykah",
14     "Fiat": "Butcher",
15     "Woo": "Erykah"
16 }
17
18 print(artist_counts(songs))
```

Output

Stack

Heap

Globals

Question 6: Function Writing Write a function definition for `count_lens` with the following expectations:

- The `count_lens` function should accept a list of string values and return a dictionary where the key type is `int` and the value type is `int`.
- The function should *count the frequencies* of strings in the parameter list of the same length(s). For example, `["a", "b", "cc", "d"]` should return `{1: 3, 2: 1}` because there were three strings of length 1 and one string of length 2.
- You should explicitly type all variables, parameters, and return types.

6.1. Write your function definition for `count_lens` here.

6.2. Write a test function for a use case that demonstrates expected usage with at least three values in the list. Your input should be different from the prompt's sample input.

Question 7: EXTRA: Want more practice with loops? Consider the following list. For each code sample below, write the corresponding output. Separate lines of output can be separated by a comma. If the code would raise an error, please write "error."

```
1 word: list[str] = ["C", "a", "t"]
```

7.1. What will be printed?

```
1 i: int = 0
2 while i < len(word):
3     print(word[i])
4     i += 1
```

7.2. What will be printed?

```
1 def grab(val: list[str]) -> str:
2     i: int = 1
3     while i < len(val):
4         return val[i]
5         i += 1
6
7 print(grab(word))
```

7.3. What will be printed?

```
1 for x in range(0, len(word)):
2     print(x)
```

7.4. What will be printed?

```
1 for x in word:
2     print(word[x])
```

7.5. What will be printed?

```
1 i: int = 0
2 while i < len(word):
3     print(word[i])
4     i += 1
```

7.6. What will be printed?

```
1 for x in range(1, len(word)):
2     print(word[x])
```

7.7. What will be printed?

```
1 for x in word:
2     print(x)
```

7.8. What will be printed?

```
1 i: int = 0
2 while i < (len(word) - 1):
3     print(i)
4     i += 1
```