Python Exercises

1. **Shopping Cart Total**

   Given a list of item prices (e.g., `[2.99, 5.49, 3.75, 10.00]`), write a loop to calculate and print the total cost.

   ```
   # Sample Solution for Q1
   prices = [2.99, 5.49, 3.75, 10.00]
   total = 0.0
   for price in prices:
       total += price

   print(f"Total cost: ${total:.2f}")
   ```

2. **Grocery List Formatter**

   You have a list of grocery items (e.g., `["apples", "bread", "milk", "eggs"]`). Use a loop to print each item on its own line with a bullet point, like:

   ```
   • apples
   • bread
   • milk
   • eggs
   ```

3. **Count Vowels in a Sentence**

   Ask the user to input a sentence (string). Loop through each character and count how many vowels (`a, e, i, o, u` in either case) are in the sentence. Print the total count.

4. **Verify PIN Code**

   Write a program that asks the user to enter a 4-digit PIN (string). Use a loop to check that each character is a digit (`'0'–'9'`). If any character isn't a digit, print "Invalid PIN," otherwise print "PIN accepted."

5. **Even-Number Filter**

   Given a list of integers representing daily steps each person took (`[1023, 4500, 6789, 2340, 1111]`), use a loop to create a new list that contains only the even numbers. Print the new list.

6. **Name Initials Extractor**

   Given a person’s full name as a single string (e.g., `"Derrick Lee Park"`), use a loop (and string methods) to extract and print their initials in uppercase (`DLP`).

7. **Temperature Conversion**

   You have a list of temperatures in Celsius (e.g., `[-5, 0, 15, 30]`). Use a loop to convert each value to Fahrenheit (`F = C * 9/5 + 32`) and store the results in a new list. Print both lists.

8. **Palindromic Words**

   Given a list of words (e.g., `["radar", "hello", "level", "world", "kayak"]`), use a loop to check which words are palindromes (same forward and backward). Print each palindrome you find.

9. **Customer Names to Uppercase**

   You have a list of customer names in mixed case (e.g., `["alice", "Bob", "cHarLie"]`). Use a loop to convert each name to all uppercase and print the results.

10. **Inventory Stock Checker**

    Given two parallel lists—`items = ["pencils", "notebooks", "erasers", "markers"]` and `stock = [12, 0, 5, 20]`—use a loop with `range` to print a message for each item: if `stock[i]` is 0, print “Out of stock: <item>,” otherwise print “<stock[i]> left of <item>.”

11. **Word Lengths in a Paragraph**

    Ask the user for a short paragraph. Split it into words and use a loop to create a list of lengths for each word (ignore punctuation). Print the list of word lengths.

12. **Remove Duplicates from a List**

    Given a list of email addresses with duplicates (e.g., `["a@x.com", "b@y.com", "a@x.com", "c@z.com"]`), use a loop to build a new list that contains each address only once. Print the de-duplicated list.

13. **Attendance List Upper/Lower Count**

    You have a list of student IDs as strings, some in uppercase (e.g., `["AB12", "cd34", "EF56", "gh78"]`). Use a loop to count how many IDs are entirely uppercase, how many are entirely lowercase, and print both counts.

14. **Count a Specific Word in a List**

    You have a list of words in a short review, for example:

```
words = ["good", "bad", "good", "neutral", "good", "bad"]
```

Ask the user to input a target word (e.g., `"good"`), then use a loop to count how many times that exact word appears in the list. Print the result (for example, “The word 'good' appears 3 times.”).

15. **Labeling Grades**

    You have a list of numerical scores (`[82, 95, 67, 70, 88]`). Use a loop to assign letter grades:

    - 90+ → "A"
    - 80–89 → "B"
    - 70–79 → "C"
    - below 70 → "D"

      Print each score alongside its letter grade (e.g., “82 → B”).

16. **Password Strength Checker (Basic)**

    Ask the user to enter a password (string). Use a loop to check and count:

    - Number of digits
    - Number of lowercase letters
    - Number of uppercase letters

      At the end, print something like: “Digits: 3, Lowercase: 5, Uppercase: 2.” (No need to enforce actual strength rules—just practice looping through the string.)

17. **Multiply List Elements by Index**

Given a list of prices (`[5.00, 10.00, 2.50, 7.75]`), use a loop with `range` to multiply each price by its index (e.g., `price[2] * 2`). Store the results in a new list and print it.

18. **Username Availability**

    Imagine `taken_usernames = ["alex", "sam", "jordan", "taylor"]`. Ask the user to enter a username (string). Loop through `taken_usernames` to see if it's already taken (case-insensitive). If it is, print "Sorry, that username is taken." Otherwise print "Username available!"

19. **Reversed Words in a Sentence**

    Ask the user for a sentence. Split it into words, then use a loop to reverse each word (e.g., "hello" → "olleh"). Finally, join and print the transformed sentence.

20. **Email Domain Extractor**

    Given a list of email addresses (e.g., `["alice@company.com", "bob@school.edu", "charlie@yahoo.com"]`), use a loop to extract the domain (everything after the "@") and store each domain in a new list. Print that list (e.g., `["company.com", "school.edu", "yahoo.com"]`).

Tips for Students

- For list-based problems, you'll often use `for item in my_list:` or `for i in range(len(my_list)):`.
- For string-based problems, remember you can treat a string like a list of characters and use slicing (e.g., `word[::-1]` to reverse).
- Test your code with small sample inputs first to make sure your loops are working before integrating the full solution.

Good luck prepping for Monday's test!