Programming with Python Feb 24-25 2025

Note: Brief Summary of contents discussed.

Revision of concepts covered so far: input, print, variable (containers for data), arithmetic operators, relational operator, logical operator, if-else, for and while loop, strings, functions: built-in and user defined, strings indexing and slicing, stings built-in functions, break and continue, lists, lists indexing and slicing, list built-in functions, list comprehension.

List Comprehension: List comprehension is a compact way to create lists. It generates a new list by applying an expression to each item in an iterable/sequence (such as list, strings, range etc) in one line.

```
Basic Syntax

new_list = [expression for item in iterable if
condition]
```

#iterable above can be range, sequence, list etc

Examples /Exercises:

1. Squaring each element

```
numbers = [1, 2, 3, 4, 5]
squared = [x**2 for x in numbers]
print(squared)
```

2. Filter even numbers

```
numbers = [1, 2, 3, 4, 5, 6]
even_numbers = [x for x in numbers if x % 2 == 0]
print(even_numbers)
# Output: [2, 4, 6]
```

3. From a given list of strings, create a new list containing only palindromic words:

```
words = ["level", "python", "radar", "world", "civic"]
# Output: ['level', 'radar', 'civic']
```

4. Given a matrix (list of lists), transpose it using list comprehension:

```
matrix = [[1, 2], [3, 4], [5, 6]]

# Output: [[1, 3, 5], [2, 4, 6]]

[[e[0] for e in matrix] for i in
range(len(matrix[0]))]
```

5. Flatten a nested list

```
nested_list = [[1, 2, 3], [4, 5], [6, 7, 8, 9]]

flattened_list = [item for sublist in nested_list for
item in sublist]

print(flattened_list)
# Output: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

6. Write a program to find the longest word in a given sequence.

7. Write a program to output an acronym for a given sequence.

For example:

```
Input: "Programming with Python"
Output: 'PWP'
```

8. Write a program to print the following pattern.

```
    *
  * * *
* * * * *
  * * *
    *
```

```
      1
    1 2 3
1 2 3 4 5
    1 2 3
      1
```