Introduction to Data Structures

Data structures are ways to store and organize data so that they can be accessed and worked with efficiently.

2. Lists

Definition and Characteristics:

- Ordered collection
- Mutable
- Allows duplicate elements

Creating and Accessing Lists:

```
# Creating a list
fruits = ['apple', 'banana', 'cherry']

# Accessing elements
print(fruits[0])  # Output: apple
print(fruits[-1]) # Output: cherry
```

Common Operations:

```
# Append an element
fruits.append('date')
print(fruits)  # Output: ['apple', 'banana', 'cherry', 'date']

# Remove an element
fruits.remove('banana')
print(fruits)  # Output: ['apple', 'cherry', 'date']

# Sort the list
fruits.sort()
print(fruits)  # Output: ['apple', 'cherry', 'date']
```

3. Tuples

Definition and Characteristics:

- Ordered collection
- Immutable
- Allows duplicate elements

Creating and Accessing Tuples:

```
# Creating a tuple colors = ('red', 'green', 'blue') # Accessing elements print(colors[1]) # Output:
green
```

4. Sets

Definition and Characteristics:

- Unordered collection
- No duplicate elements

Creating and Accessing Sets:

```
# Creating a set

numbers = {1, 2, 3, 4, 5}

[1,2,3,4,5,5]

len(num)

[“dog”, “cat”, “cow”]


# Accessing elements (using a loop since sets are unordered)

for num in numbers:

    print(num)


# Union of sets

set1 = {1, 2, 3}

set2 = {3, 4, 5}
```

```
print(set1 | set2)  # Output: {1, 2, 3, 4, 5}


# Intersection of sets

print(set1 & set2)  # Output: {3}


# Difference of sets

print(set1 - set2)  # Output: {1, 2}
```

5. Dictionaries

Definition and Characteristics:

- Unordered collection
- Key-value pairs
- Keys must be unique

Creating and Accessing Dictionaries:

```
# Creating a dictionary

student_scores = {'Alice': 85, 'Bob': 90, 'Charlie': 78}


# Accessing elements

print(student_scores['Alice'])  # Output: 85


# Add a new key-value pair

student_scores['David'] = 92

print(student_scores)  # Output: {'Alice': 85, 'Bob': 90, 'Charlie': 78, 'David': 92}


# Remove a key-value pair
```

```
del student_scores['Charlie']

print(student_scores)  # Output: {'Alice': 85, 'Bob': 90, 'David': 92}


# Get all keys and values

print(student_scores.keys())  # Output: dict_keys(['Alice', 'Bob', 'David'])

print(student_scores.values())  # Output: dict_values([85, 90, 92])
```

Practice question:

1. List

- Create a list of numbers from 1 to 10.
- Write a program to find the maximum and minimum elements in a list.

2. Tuples

- Create a tuple with different data types (int, float, string).
- Write a program to concatenate two tuples.

3. Sets

- Create a set of prime numbers less than 10.
- Write a program to find the common elements between two sets.

4. Dictionaries

- Create a dictionary with the names and ages of your friends.
- Write a program to update the score of a student and print the updated dictionary.

Python for Machine Learning -
https://www.youtube.com/watch?v=7eh4d6sabA0

Numpy -
https://www.youtube.com/watch?v=Rbh1rieb3zc

Pandas -
https://www.youtube.com/watch?v=RhEjmHeDNoA