ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

TRAINING TR-102 REPORT DAY 8 2 JULY 2025

Overview:

The eighth day of training focused on strengthening our Python programming skills by exploring core concepts such as string manipulation, lists, tuples, dictionaries, sets, loops, and functions.We also implemented a basic calculator program to apply all the concepts we learned.This session helped us understand how to handle data structures efficiently, perform iterations, and create reusable code using functions — all of which are essential skills in AI and Machine Learning programming.

Learning Objectives:

- Understand string operations like slicing and concatenation.
- Learn Python data structures — list, tuple, dictionary, and set.
- Explore loops and their applications in programming.
- Learn how to define and call functions.
- Apply all concepts to build a simple calculator program.

Python Strings:

Strings are sequences of characters enclosed in quotes (' ' or " ").
They are immutable, meaning their content cannot be changed after creation.

Example:

```
name = "Artificial Intelligence"
```

Python Lists:

A list is an ordered, mutable collection of elements.

Example:

```
fruits = ["apple", "banana", "cherry"]

print(fruits[1])      # Output: banana
```

Python Tuples

A tuple is similar to a list but immutable (cannot be changed).

Example:

```
colors = ("red", "green", "blue")

print(colors[0])   # Output: red
```

Python Dictionaries

A **dictionary** stores data in key-value pairs.

Example:

```
student = {"name": "Aditi", "age": 21, "course": "AI"}

print(student["name"])   # Output: Aditi
```

Python Sets:

A set is an unordered collection of unique elements.

Example:

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

print(numbers)    # Output: {1, 2, 3, 4}
```

Loops in Python:

Loops are used to execute a block of code repeatedly.

For Loop Example:

```
for i in range(5):
    print("Iteration:", i)
```

While Loop Example:

```
count = 0
while count < 3:
    print("Count:", count)
    count += 1
```

Functions in Python

Functions are reusable blocks of code that perform a specific task.

Defining and Calling a Function:

```
     def greet(name):
         print("Hello,", name)
greet("Aditi")
```

Function with Return Value:

```
def add(a, b):
return a + b
result = add(5, 10)
print(result)   # Output: 15
```

Calculator Program

```
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    if y != 0:
        return x / y
    else:
        return "Cannot divide by zero"

print("Select Operation:")
print("1. Add\n2. Subtract\n3. Multiply\n4. Divide")

choice = input("Enter choice (1/2/3/4): ")
```

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
```

```
    if choice == '1':
    print("Result:", add(num1, num2))

elif choice == '2':

    print("Result:", subtract(num1, num2))

elif choice == '3':

    print("Result:", multiply(num1, num2))

elif choice == '4':

    print("Result:", divide(num1, num2))

else:

    print("Invalid choice!")
```

Output

```
Select Operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 3
Enter first number: 5
Enter second number: 8
Result: 40.0

=== Code Execution Successful ===
```

Conclusion:

Day 8 provided an in-depth understanding of Python's core programming structures and practical coding skills. We learned to manipulate strings and collections, apply loops for iteration, and modularize code using functions.

By implementing a calculator program, we applied all these concepts in a hands-on way, enhancing our confidence in Python programming.This session laid a strong foundation for developing data-driven applications and AI algorithms using Python.