



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



Experiment No.1: Programs on basic control structures & loops.

1)Conditional (if-else) Statement:

Write a program that determines if a given number is even or odd.

```
# Program to determine if a number is even or odd num = int(input("Enter a number:"))
if num % 2 == 0: print(f"{num} is even.")
else:
print(f"{num} is odd.")
```

2)While Loop:

Write a program that prints numbers from 1 to 5 using a while loop.

```
# Program to print numbers from 1 to 5 using a while loop count = 1
while count <= 5: print(count) count += 1
```

3)For Loop:

Write a program that calculates the sum of numbers from 1 to 10 using a for loop.

```
# Program to calculate the sum of numbers from 1 to 10 using a for loop sum = 0
for i in range(1, 11): sum += i
print("Sum of numbers from 1 to 10 is:", sum)
```

4)Nested Loop:

Write a program that prints a pattern of stars using nested loops.

```
# Program to print a pattern of stars in a nested loop n = 5
for i in range(1, n + 1): for j in range(i):
print("*", end="") print()
```

5)Break and Continue:

Write a program that finds the first even number in a list and skips the rest if the number is odd.

```
# Program to find the first even number in a list and continue to the next number if odd
numbers = [7, 12, 5, 8, 3, 10] for num in numbers:
if num % 2 == 0:
print(f"The first even number in the list is {num}.") break
else:
Continue
```

6)Switch Case (Using Dictionary):

Write a program that performs basic arithmetic operations (addition, subtraction, multiplication, division) using a dictionary to simulate a switch case.



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
# Program to implement a simple switch case using a dictionary 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): return x / y
```

```
choice = input("Enter operation (+, -, *, /):") num1 = float(input("Enter first number:")) num2 =  
float(input("Enter second number:"))
```

```
operations = {  
'+' : add(num1, num2),  
 '-' : subtract(num1, num2),  
 '*' : multiply(num1, num2),  
 '/' : divide(num1, num2)  
}
```

```
if choice in operations:  
    print(f"Result: {operations[choice]}") else:  
    print("Invalid operation")
```

Experiment No.2: Programs on operators & I/O operations.

1)Arithmetic Operators:

Write a program to perform arithmetic operations on two numbers.

```
# Program to perform arithmetic operations on two numbers num1 = float(input("Enter the  
first number: "))  
num2 = float(input("Enter the second number: "))
```

```
sum = num1 + num2 difference = num1 - num2 product = num1 * num2 quotient = num1 /  
num2  
print(f"Sum: {sum}") print(f"Difference: {difference}") print(f"Product: {product}")  
print(f"Quotient: {quotient}")
```

2)Comparison Operators:

Write a program to compare two numbers.

```
# Program to compare two numbers
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



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

```
if num1 > num2:
print(f"{num1} is greater than {num2}") elif num1 < num2:
print(f"{num1} is less than {num2}") else:
print(f"{num1} is equal to {num2}")
```

3)Logical Operators:

Write a program to check if a number is within a specified range.

```
# Program to check if a number is within a specified range num = int(input("Enter a number:
"))
lower_limit = 10
upper_limit = 50
```

```
if num >= lower_limit and num <= upper_limit:
print(f"{num} is within the range [{lower_limit}, {upper_limit}]") else:
print(f"{num} is outside the range.")
```

4)Input/Output Operations:

Write a program to read and write to a text file.

```
# Program to read and write to a text file file_name = "sample.txt"
```

```
# Writing to a file
```

```
with open(file_name, 'w') as file: file.write("Hello, this is a sample file.\n")
file.write("Python is a great programming language.")
```

```
# Reading from a file
```

```
with open(file_name, 'r') as file: content = file.read() print("File Contents:") print(content)
```

5)String Operators:

Write a program to concatenate two strings entered by the user.

```
# Program to concatenate strings str1 = "Hello, "
str2 = "world!"
```

```
concatenated_str = str1 + str2 print(concatenated_str)
```

```
# Program to repeat a string original_str = "Python"
repeated_str = original_str * 3 print(repeated_str)
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



Experiment No.3: Programs on Lists.

1)Creating and Accessing Lists:

Create a list of fruits and access different elements of the list.

Program to create a list and access its elements

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

Accessing elements

```
print("First fruit:", fruits[0]) print("Last fruit:", fruits[-1])
```

```
print("Slicing the list:", fruits[1:3])
```

2)Modifying Lists:

Modify a list of numbers by appending, updating, and removing elements.

Program to modify a list

```
numbers = [1, 2, 3, 4, 5]
```

Appending an element

```
numbers.append(6)
```

Updating an element

```
numbers[2] = 10
```

Removing an element

```
numbers.remove(4)
```

Printing the modified list

```
print(numbers)
```

3)Iterating Over Lists:

Iterate over a list of fruits and print each fruit.

Program to iterate over a list

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

```
for fruit in fruits:
```

```
print(fruit)
```

4)List Comprehension:

Create a new list of squared numbers using list comprehension.

Program to create a new list using list comprehension numbers = [1, 2, 3, 4, 5]



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
squared_numbers = [num**2 for num in numbers] print(squared_numbers)
```

5)Finding Elements in Lists:

Check if a specific fruit is in the list and print a corresponding message.

Program to find an element in a list

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

```
if search_fruit in fruits: print(f"{search_fruit} is in the list.")
```

```
else:
```

```
    print(f"{search_fruit} is not in the list.")
```

6)Sorting and Reversing Lists:

Sort a list of numbers in ascending order and then reverse it.

Program to sort and reverse a list

```
numbers = [3, 1, 4, 2, 5]
```

Sorting the list in ascending order

```
numbers.sort()
```

```
print("Sorted list:", numbers)
```

Reversing the list

```
numbers.reverse() print("Reversed list:", numbers)
```

Experiment No.4: Programs on Tuples.

1)Creating and Accessing Tuples:

Write a program to create a tuple and access its elements using indexing and slicing.

Program to create a tuple and access its elements

```
fruits = ("apple", "banana", "cherry", "date")
```

Accessing elements

```
print("First fruit:", fruits[0]) print("Last fruit:", fruits[-1])
```

```
print("Slicing the tuple:", fruits[1:3])
```

2)Tuple Packing and Unpacking:

Write a program to demonstrate tuple packing and unpacking.

Program to use tuple packing and unpacking person = ("John", 30, "Engineer")

Unpacking the tuple into variables name, age, occupation = person

```
print(f"Name: {name}")
```

```
print(f"Age: {age}") print(f"Occupation: {occupation}")
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



3)Combining Tuples:

Write a program to combine two tuples into one.

```
# Program to combine two tuples fruits = ("apple", "banana") more_fruits = ("cherry", "date")
```

```
combined_fruits = fruits + more_fruits
```

```
print("Combined tuple:", combined_fruits)
```

4)Finding Elements in Tuples:

Write a program to check if a specific element exists in a tuple.

```
# Program to find an element in a tuple fruits = ("apple", "banana", "cherry", "date")
```

```
search_fruit = "banana"
```

```
if search_fruit in fruits: print(f"{search_fruit} is in the tuple.")
```

```
else:
```

```
print(f"{search_fruit} is not in the tuple.")
```

5)Counting and Indexing:

Write a program to count the occurrences of a specific element and find the index of an element in a tuple.

```
# Program to count occurrences and find the index of an element in a tuple fruits = ("apple", "banana", "cherry", "banana", "date")
```

```
count = fruits.count("banana") index = fruits.index("cherry")
```

```
print(f"Count of 'banana': {count}") print(f"Index of 'cherry': {index}")
```

Experiment No.5: Programs on Dictionary.

1)Creating and Accessing Dictionaries:

Write a program to create a dictionary and access its elements using keys.

```
# Program to create a dictionary and access its elements
```

```
student = {  
    "name": "Alice", "roll_number": 101,  
    "marks": 95  
}
```

```
# Accessing elements
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
print("Student Name:", student["name"]) print("Roll Number:", student["roll_number"])
print("Marks:", student["marks"])
```

2)Modifying Dictionaries:

Write a program to modify an existing dictionary by updating, adding, and removing key-value pairs.

Program to modify a dictionary

```
student = {
"name": "Alice", "roll_number": 101,
"marks": 95
}
```

Updating a value

```
student["marks"] = 98
```

Adding a new key-value pair

```
student["grade"] = "A"
```

Removing a key-value pair del student["roll_number"]

Printing the modified dictionary print("Modified Student Info:", student)

3)Iterating Over Dictionary:

Write a program to iterate over a dictionary and print each key-value pair.

Program to iterate over a dictionary

```
student = {
"name": "Alice", "roll_number": 101,
"marks": 95
}
```

```
for key, value in student.items():
```

```
    print(f'{key}: {value}')
```

4)Checking for Key Existence:

Write a program to check if a specific key exists in a dictionary.

Program to check if a key exists in a dictionary student = {

```
"name": "Alice", "roll_number": 101,
"marks": 95
}
```

```
search_key = "grade"
```

```
if search_key in student:
```

```
    print(f'{search_key} exists in the dictionary.') else:
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
print(f"{search_key} does not exist in the dictionary.")
```

5) Dictionary Comprehension:

Write a program to create a new dictionary where the keys are numbers and the values are the squares of those numbers.

```
# Program to create a new dictionary using dictionary comprehension
```

```
numbers = [1, 2, 3, 4, 5]
```

```
squared_dict = {num: num**2 for num in numbers} print("Squared Dictionary:", squared_dict)
```

Experiment No.6: Programs on Set.

1. Creating and Accessing Sets:

```
# Program to create a set and access its elements
```

```
fruits = {"apple", "banana", "cherry", "date"}
```

```
# Accessing elements
```

```
for fruit in fruits:
```

```
    print(fruit)
```

2. Modifying Sets:

```
# Program to modify a set
```

```
fruits = {"apple", "banana", "cherry", "date"}
```

```
# Adding an element
```

```
fruits.add("grape")
```

```
# Removing an element
```

```
fruits.remove("cherry")
```

```
# Printing the modified set
```

```
print("Modified Set:", fruits)
```

3. Set Operations:

```
# Program to perform set operations
```

```
A = {1, 2, 3, 4, 5}
```

```
B = {3, 4, 5, 6, 7}
```

```
# Union of sets
```

```
union = A | B
```

```
print("Union of A and B:", union)
```

```
# Intersection of sets
```




Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
intersection = A & B  
print("Intersection of A and B:", intersection)
```

```
# Difference of sets  
difference = A - B  
print("Difference of A and B:", difference)
```

4. Set Comprehension:

```
# Program to create a new set using set comprehension  
numbers = {1, 2, 3, 4, 5}
```

```
squared_set = {num**2 for num in numbers}  
print("Squared Set:", squared_set)
```

Experiment No.7: Programs on File Handling.

1. Writing to a Text File:

```
# Program to write data to a text file  
file_name = "sample.txt"  
  
# Open the file in write mode (creates a new file or overwrites an existing one)  
with open(file_name, 'w') as file:  
    file.write("Hello, this is a sample text file.\n")  
    file.write("Python is a great programming language.")  
  
print(f"Data written to '{file_name}' successfully.")
```

2. Reading from a Text File:

```
# Program to read data from a text file  
file_name = "sample.txt"  
  
# Open the file in read mode  
with open(file_name, 'r') as file:  
    content = file.read()  
  
print("File Contents:")  
print(content)
```

3. Appending to a Text File:

```
# Program to append data to an existing text file  
file_name = "sample.txt"  
  
# Open the file in append mode (keeps the existing content and adds new content)
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
with open(file_name, 'a') as file:  
file.write("\nThis is an additional line added to the file.")  
  
print("Data appended to the file successfully.")
```

Experiment No.8: Programs on Strings.

1. String Concatenation:

```
# Program to concatenate two strings  
str1 = "Hello, "  
str2 = "world!"
```

```
concatenated_str = str1 + str2  
print(concatenated_str)
```

2. String Length:

```
# Program to find the length of a string  
text = "Python is a powerful language."
```

```
length = len(text)  
print(f"The length of the string is {length} characters.")
```

3. String Slicing:

```
# Program to slice a string  
text = "Python is fun!"
```

```
# Slicing the string to get a substring  
substring = text[7:9]  
print(f"Substring: {substring}")
```

4. String Methods:

```
# Program to use string methods  
text = "Hello, world!"
```

```
# Convert to uppercase  
uppercase_text = text.upper()  
print("Uppercase:", uppercase_text)
```

```
# Convert to lowercase  
lowercase_text = text.lower()  
print("Lowercase:", lowercase_text)
```

```
# Replace a substring
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
replaced_text = text.replace("world", "Python")  
print("Replaced Text:", replaced_text)
```

5. String Splitting:

```
# Program to split a string into a list  
text = "apple, banana, cherry, date"
```

```
# Split the string by commas  
fruits_list = text.split(", ")  
print("List of Fruits:", fruits_list)
```

Experiment 9: Programs on NumPy Operations

1)Array Creation:

```
# Program to create a 1D and 2D array using NumPy  
import numpy as np
```

```
# Creating a 1D array  
array_1d = np.array([1, 2, 3, 4, 5])  
print("1D Array:", array_1d)
```

```
# Creating a 2D array  
array_2d = np.array([[1, 2, 3], [4, 5, 6]])  
print("2D Array:\n", array_2d)
```

2)Array Shape and Size:

```
# Program to find shape and size of a NumPy array  
import numpy as np
```

```
array = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

```
# Getting the shape of the array  
print("Shape of array:", array.shape)
```

```
# Getting the size (number of elements)  
print("Size of array:", array.size)
```

3)Array Slicing:

```
# Program to slice a NumPy array  
import numpy as np
```



Lokmanya Tilak Jankalyan Shikshan Sanstha's
PRIYADARSHINI COLLEGE OF ENGINEERING

(Recognised by A.I.C.T.E., New Delhi & Govt. of Maharashtra, Affiliated to R.T.M.Nagpur University)
Near CRPF Campus, Hingna Road, Nagpur-440 019, Maharashtra (India)

Department of Robotics & Artificial Intelligence

2024-25 (Odd Sem)



```
array = np.array([10, 20, 30, 40, 50])
```

```
# Slicing the array to get elements from index 1 to 3
```

```
sliced_array = array[1:4]
```

```
print("Sliced Array:", sliced_array)
```

4)Array Reshaping:

```
# Program to reshape a NumPy array
```

```
import numpy as np
```

```
array = np.array([1, 2, 3, 4, 5, 6])
```

```
# Reshaping into a 2x3 array
```

```
reshaped_array = array.reshape(2, 3)
```

```
print("Reshaped Array:\n", reshaped_array)
```

5)Array Arithmetic Operations:

```
# Program for arithmetic operations on NumPy arrays
```

```
import numpy as np
```

```
array1 = np.array([1, 2, 3])
```

```
array2 = np.array([4, 5, 6])
```

```
# Element-wise addition
```

```
addition = array1 + array2
```

```
print("Addition:", addition)
```

```
# Element-wise multiplication
```

```
multiplication = array1 * array2
```

```
print("Multiplication:", multiplication)
```

6)Array Broadcasting:

```
# Program to demonstrate broadcasting in NumPy
```

```
import numpy as np
```

```
array = np.array([1, 2, 3])
```

```
# Broadcasting a scalar value
```

```
broadcasted_array = array + 10
```

```
print("Broadcasted Array:", broadcasted_array)
```