ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University

(SEWA)

SKILL ENHANCEMENT **EMPLOYABILITY** **WISDOM** **ACCESSIBILITY**

Bachelor of Computer Applications (BCA)

Course : Python Programming Lab

Course Code: BCA-2-02P

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PROGRAMME COORDINATOR :
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

PROGRAMME CO-COORDINATOR :
Dr. Gaurav Dhiman
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE COORDINATOR :
Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,

Dean Academic Affairs

Bachelor of Computer Applications (BCA)
BCA-2-02P: Python Programming Lab

Total Marks: 50
External Marks: 35
Internal Marks: 15
Credits: 2
Pass Percentage: 40%

Course: Python Programming Lab	
Course Code:BCA-2-02P	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Demonstrate proficiency in writing Python code to solve simple problems.
CO2	Use and manipulate basic data structures in Python, such as lists, tuples, and dictionaries.
CO3	Solve algorithmic problems using Python.
CO4	Utilize common Python libraries for specific tasks (e.g., NumPy for numerical computing, Pandas for data manipulation).
CO5	Use libraries for data manipulation, analysis, and visualization.

Detailed List of Programmes:

Programme No.	**Name of Programme**
P1	WAP to find the sum of two numbers
P2	WAP to find Area of Rectangle and Circle
P3	WAP to find Volume of Sphere
P4	WAP to find the maximum of three numbers in python
P5	WAP to print all the prime numbers between two numbers
P6	WAP to print FIBONACCI SERIES using WHILE Loop
P7	WAP to print FIBONACCI SERIES using FOR Loop
P8	WAP to calculate X^n by FOR Loop

P9	WAP to print FACTORIAL of List
P10	WAP to create a list of values inputted by user and sort in increasing order.
P11	WAP to find given number is PRIME or not.
P12	Demonstrate the use of polymorphism by creating different functions for sum.
P13	WAP to write an exception for divisibility of a number by 0.
P14	WAP to print the first 10 lines in text file using python
P15	Take a list of integers. WAP to find the pairs which give the minimum difference
P16	WAP to copy content from a file in a computer and paste it into another file.
P17	WAP to create a linked list using python
P18	Demonstrate the use of common Python libraries for specific tasks (e.g., NumPy for numerical computing, Pandas for data manipulation)
P19	Demonstrate the use of common Python libraries for data manipulation, analysis, and visualization.

Bachelor of Computer Applications (BCA)
Python Programming Lab

1. **WAP to find the sum of two numbers**

```
def sum_of_two_numbers(a, b):
    return a + b

# Example usage
num1 = 5
num2 = 7
result = sum_of_two_numbers(num1, num2)
print(f"The sum of {num1} and {num2} is {result}")
```

2. **WAP to find Area of Rectangle and Circle**

```
import math

# Function to find the area of a rectangle
def area_of_rectangle(length, width):
    return length + width

# Function to find the area of a circle
def area_of_circle(radius):
    return math.pi * radius * radius

# Example usage
length = 10
width = 5
radius = 7
area_rectangle = area_of_rectangle(length, width)
area_circle = area_of_circle(radius)

print(f"The area of the rectangle is {area_rectangle}")
print(f"The area of the circle is {area_circle}")
```

3. **WAP to find Volume of Sphere**

```
import math

# Function to find the volume of a sphere
def volume_of_sphere(radius):
    return (4/3) * math.pi * radius ** 3

# Example usage
```

```
radius = 5
volume = volume_of_sphere(radius)
print(f"The volume of the sphere with radius {radius} is {volume}")
```

4. **WAP to find the maximum of three numbers in python**

```
# Function to find the maximum of three numbers
def max_of_three_numbers(a, b, c):
    return max(a, b, c)

# Example usage
num1 = 10
num2 = 20
num3 = 15
maximum = max_of_three_numbers(num1, num2, num3)
print(f"The maximum of {num1}, {num2}, and {num3} is {maximum}")
```

5. **WAP to print all the prime numbers between two numbers**

```
# Function to check if a number is prime
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# Function to print all prime numbers between two numbers
def prime_numbers_between(start, end):
    primes = []
    for num in range(start, end + 1):
        if is_prime(num):
            primes.append(num)
    return primes

# Example usage
start = 10
end = 50
primes = prime_numbers_between(start, end)
print(f"Prime numbers between {start} and {end} are: {primes}")
```

6. **WAP to print FIBONACCI SERIES using WHILE Loop**

```
# Function to print Fibonacci series using WHILE loop
def fibonacci_while(n):
```

```
    a, b = 0, 1
    count = 0
    while count < n:
        print(a, end=" ")
        a, b = b, a + b
        count += 1

# Example usage
n = 10
fibonacci_while(n)
```

7. WAP to print FIBONACCI SERIES using FOR Loop

```
# Function to print Fibonacci series using FOR loop
def fibonacci_for(n):
    a, b = 0, 1
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b

# Example usage
n = 10
fibonacci_for(n)
```

8. WAP to calculate X^n by FOR Loop

```
# Function to calculate X^n using FOR loop
def power(x, n):
    result = 1
    for _ in range(n):
        result *= x
    return result

# Example usage
x = 2
n = 3
print(f"{x}^{n} is {power(x, n)}")
```

9. WAP to print FACTORIAL of List

```
# Function to calculate factorial of a number
def factorial(n):
    if n == 0:
        return 1
    result = 1
    for i in range(1, n + 1):
```

```
        result *= i
    return result

# Function to print factorial of a list of numbers
def factorial_of_list(numbers):
    return [factorial(n) for n in numbers]

# Example usage
numbers = [3, 4, 5]
print(f"Factorials of {numbers} are {factorial_of_list(numbers)}")
```

10. WAP to create a list of values inputted by user and sort in increasing order.

```
# Function to create and sort a list of values inputted by user
def create_and_sort_list():
    user_input = input("Enter numbers separated by spaces: ")
    numbers = list(map(int, user_input.split()))
    numbers.sort()
    return numbers

# Example usage
sorted_numbers = create_and_sort_list()
print(f"Sorted list: {sorted_numbers}")
```

11. WAP to find given number is PRIME or not.

```
# Function to check if a number is prime
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# Example usage
number = 29
if is_prime(number):
    print(f"{number} is a prime number")
else:
    print(f"{number} is not a prime number")
```

12. Demonstrate the use of polymorphism by creating different functions for sum.

```
# Polymorphic functions for sum
```

```
# Function to sum two integers
def sum(a, b):
    return a + b

# Function to sum a list of numbers
def sum(numbers):
    return sum(numbers)

# Example usage
num1, num2 = 5, 10
numbers = [1, 2, 3, 4, 5]
print(f"Sum of {num1} and {num2} is {sum([num1, num2])}")
print(f"Sum of the list {numbers} is {sum(numbers)}")
```

13. WAP to write an exception for divisibility of a number by 0.

```
# Function to handle division and catch division by zero exception
def divide(a, b):
    try:
        result = a / b
        return result
    except ZeroDivisionError:
        return "Error: Division by zero is not allowed."

# Example usage
a, b = 10, 0
print(divide(a, b))
```

14. WAP to print the first 10 lines in text file using python

```
# Function to print first 10 lines of a text file
def print_first_10_lines(file_path):
    with open(file_path, 'r') as file:
        for i in range(10):
            line = file.readline()
            if line == "":
                break
            print(line, end="")

# Example usage
file_path = 'example.txt'
print_first_10_lines(file_path)
```

15. Take a list of integers. WAP to find the pairs which give the minimum difference

```
# Function to find pairs with minimum difference
```

```
def find_min_diff_pairs(numbers):
    numbers.sort()
    min_diff = float('inf')
    pairs = []
    for i in range(len(numbers) - 1):
        diff = numbers[i + 1] - numbers[i]
        if diff < min_diff:
            min_diff = diff
            pairs = [(numbers[i], numbers[i + 1])]
        elif diff == min_diff:
            pairs.append((numbers[i], numbers[i + 1]))
    return pairs

# Example usage
numbers = [4, 2, 1, 7, 5]
print(f"Pairs with minimum difference: {find_min_diff_pairs(numbers)}")
```

16. WAP to copy content from a file in a computer and paste it into another file.

```
# Function to copy content from one file to another
def copy_file(source_file, destination_file):
    with open(source_file, 'r') as src:
        content = src.read()
    with open(destination_file, 'w') as dst:
        dst.write(content)

# Example usage
source_file = 'source.txt'
destination_file = 'destination.txt'
copy_file(source_file, destination_file)
```

17. WAP to create a linked list using python

```
# Node class
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

# Linked list class
class LinkedList:
    def __init__(self):
        self.head = None

    def append(self, data):
        new_node = Node(data)
```

```
        if not self.head:
            self.head = new_node
        else:
            last_node = self.head
            while last_node.next:
                last_node = last_node.next
            last_node.next = new_node

    def display(self):
        current = self.head
        while current:
            print(current.data, end=" -> ")
            current = current.next
        print("None")

# Example usage
ll = LinkedList()
ll.append(1)
ll.append(2)
ll.append(3)
ll.display()
```

18. Demonstrate the use of common Python libraries for specific tasks (e.g., NumPy for numerical computing, Pandas for data manipulation)

```
# Using NumPy for numerical computing
import numpy as np

# Example: Creating an array and performing an operation
array = np.array([1, 2, 3, 4, 5])
print(f"NumPy array: {array}")
print(f"Array doubled: {array * 2}")

# Using Pandas for data manipulation
import pandas as pd

# Example: Creating a DataFrame and performing an operation
data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}
df = pd.DataFrame(data)
print("Pandas DataFrame:")
print(df)
print("DataFrame after adding a column:")
df['Score'] = [85, 90, 95]
print(df)
```

19. Demonstrate the use of common Python libraries for data manipulation, analysis, and visualization.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Data manipulation with Pandas
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'], 'Age': [25, 30, 35, 40], 'Score': [85, 88, 92, 95]}
df = pd.DataFrame(data)
print("Pandas DataFrame:")
print(df)

# Data analysis with NumPy
mean_age = np.mean(df['Age'])
print(f"Mean age: {mean_age}")

# Data visualization with Matplotlib
plt.figure(figsize=(8, 5))
plt.bar(df['Name'], df['Score'], color='blue')
plt.xlabel('Name')
plt.ylabel('Score')
plt.title('Scores by Name')
plt.show()

# Data visualization with Seaborn
sns.set(style="whitegrid")
sns.barplot(x='Name', y='Score', data=df)
plt.title('Scores by Name (Seaborn)')
plt.show()
```