**Python Programming (28521)**

**Experiment No:** 01 **Date:** / /

**Name of Experiment:** Write & execute programs using variables & operators.

**Necessary Instruments:**

* Personal Computer (Laptop/Desktop)
* Python Software
* Pycharm IDE
* Internet Connection

**Objective:** To gain knowledge about executing programs using variables & operators.

**Working Procedure:**

Program No: 01

**Name of Program:** Write a Python program to display the sentence “Hello World” on the monitor screen.

**Code:**

print(“Hello World”)

............................................................................................................................... **Output:**

Hello World

...............................................................................................................................

Program No: 02

**Name of Program:** Write a Python program to calculate addition, subtraction, multiplication and division between two numbers.

**Code:**

a = int(input(“Enter the first number : ”))

b = int(input(“Enter the second number : ”))

sum = a + b

sub = a – b

mul = a \* b

div = a / b

print(“A + B = ”, sum)

print(“A - B = ”, sub)

print(“A \* B = ”, mul)

print(“A / B = ”, div)

............................................................................................................................... **Output:**

Enter the first number : 4

Enter the second number : 2

A + B = 6

A - B = 2

A \* B = 8

A / B = 2

...............................................................................................................................

**Carefulness:** Ensure input validation for numerical values, maintain correct syntax and indentation, and define and call functions appropriately. Verify mathematical operations and logical conditions for accuracy.

**Remark:** Both programs effectively use variables and operators to handle conditions based on user input, providing clear prompts and correct outputs.

Course Teacher Head of the Dept. Principal

**Python Programming (28521)**

**Experiment No:** 02 **Date:** / /

**Name of Experiment:** Write & execute programs using branching statements.

**Necessary Instruments:**

* Personal Computer (Laptop/Desktop)
* Python Software
* Pycharm IDE
* Internet Connection

**Objective:** To gain knowledge about executing programs using branching statements.

**Working Procedure:**

Program No: 01

**Name of Program:** Write a Python program to find the biggest number from three numbers.

**Code:**

def max3val():

a = int(input(“Enter first number =”))

b = int(input(“Enter second number =”))

c = int(input(“Enter third number =”))

if (a>b) and (a>c):

print(“A is maximum and it is :”, a)

elif (b>c):

print(“B is maximum and it is :”, b)

else:

print(“C is maximum and it is :”, c)

max3val()

............................................................................................................................... **Output:**

Enter first number = 81

Enter second number = 25

Enter third number = 59

A is maximum and it is : 81

...............................................................................................................................

Program No: 02

**Name of Program:** Write a Python program to calculate triangle area using function.

**Code:**

import math

def triangle\_area():

a = float(input(“Enter first arm length=”))

b = float(input(“Enter second arm length =”))

c = float(input(“Enter third arm length =”))

if (a+b)>c and (b+c)>a and (a+c)>b:

s = (a+b+c)/2

area = math.sqrt(s\*(s-a)\*(s-b)\*(s-c))

print(“Triangle area is = ”, area)

else:

print(“Triangle is not possible”)

triangle\_area()

............................................................................................................................... **Output:**

Enter first arm length = 7

Enter second arm length = 9

Enter third arm length = 6

Triangle area is = 20.97617696340303

...............................................................................................................................

**Carefulness:** Ensure input validation for numerical values, maintain correct syntax and indentation, and define and call functions appropriately. Verify mathematical operations and logical conditions for accuracy.

**Remark:** Both programs effectively use branching statements to handle conditions based on user input, providing clear prompts and correct outputs.

Course Teacher Head of the Dept. Principal

**Python Programming (28521)**

**Experiment No:** 03 **Date:** / /

**Name of Experiment:** Write & execute programs using looping statements.

**Necessary Instruments:**

* Personal Computer (Laptop/Desktop)
* Python Software
* Pycharm IDE
* Internet Connection

**Objective:** To gain knowledge about executing programs using looping statements.

**Working Procedure:**

Program No: 01

**Name of Program:** Write a Python program to print even numbers from 1 to n.

**Code:**

n = int(input(“Enter the end of Range = ”))

print(“Printing of Even Number Series:”)

for i in range(1, n+1):

if not(i%2):

print(i, end=“ ”)

............................................................................................................................... **Output:**

Enter the end of Range = 50

Printing of Even Number Series:  
 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

...............................................................................................................................

Program No: 02

**Name of Program:** Write a Python program to print prime numbers from 1 to n.

**Code:**

lower\_value = int(input(“Enter the lowest range value : ”))

upper\_value = int(input(“Enter the upper range value : ”))

print(“The prime numbers in the range are : ”)

for number in range (lower\_value, upper\_value+1):

if number > 1:

for i in range(2, number):

if (number % i) == 0:

break

else:

print(number, “ ”, end=“ ”)

............................................................................................................................... **Output:**

Enter the lowest range value: 9

Enter the upper range value: 66

The prime numbers in the range are :

11 13 17 19 23 29 31 37 41 43 47 53 59 61 ...............................................................................................................................

**Carefulness:** Ensure input validation for numerical values, maintain correct syntax and indentation, and define and call functions appropriately. Verify mathematical operations and logical conditions for accuracy.

**Remark:** Both programs effectively use looping statements to handle conditions based on user input, providing clear prompts and correct outputs.

Course Teacher Head of the Dept. Principal

**Python Programming (28521)**

**Experiment No:** 04 **Date:** / /

**Name of Experiment:** Write & execute programs using lists/array.

**Necessary Instruments:**

* Personal Computer (Laptop/Desktop)
* Python Software
* Pycharm IDE
* Internet Connection

**Objective:** To gain knowledge about executing programs using lists/array.

**Working Procedure:**

Program No: 01

**Name of Program:** Write a Python program to take input in lists and find smallest and largest numbers from the lists using function.

**Code:**

def find\_min\_max():

n = int(input("Enter the number of elements in the list: "))

lst = []

for i in range(n):

elem = int(input(f"Enter element {i+1}: "))

lst.append(elem)

smallest = min(lst)

largest = max(lst)

print(f"The smallest number is: {smallest}")

print(f"The largest number is: {largest}")

find\_min\_max()

............................................................................................................................... **Output:**

Enter the number of elements in the list: 5

Enter element 1: 12

Enter element 2: 4

Enter element 3: 19

Enter element 4: 1

The smallest number is: 1

The largest number is: 19 ...............................................................................................................................

Program No: 02

**Name of Program:** Write a Python program to calculate the sum of the elements in a list.

**Code:**

def sum\_of\_list():

n = int(input("Enter the number of elements in the list: "))

lst = []

for i in range(n):

elem = int(input(f"Enter element {i+1}: "))

lst.append(elem)

total\_sum = sum(lst)

print(f"The sum of the elements in the list is: {total\_sum}")

sum\_of\_list()

............................................................................................................................... **Output:**

Enter the number of elements in the list: 5

Enter element 1: 12

Enter element 2: 4

Enter element 3: 19

Enter element 4: 1

Enter element 5: 8

The sum of the elements in the list is: 44

...............................................................................................................................

**Carefulness:** Ensure input validation for numerical values, maintain correct syntax and indentation, and define and call functions appropriately. Verify mathematical operations and logical conditions for accuracy.

**Remark:** Both programs effectively use lists operation to handle conditions based on user input, providing clear prompts and correct outputs.

Course Teacher Head of the Dept. Principal

**Python Programming (28521)**

**Experiment No:** 05 **Date:** / /

**Name of Experiment:** Write & execute programs using a dictionary.

**Necessary Instruments:**

* Personal Computer (Laptop/Desktop)
* Python Software
* Pycharm IDE
* Internet Connection

**Objective:** To gain knowledge about executing programs using dictionary.

**Working Procedure:**

Program No: 01

**Name of Program:** Write a Python program to create a empty dictionary.

**Code:**

my\_dict\_one = {}

my\_dict\_two = {}

print(my\_dict\_one)

print(my\_dict\_two)

............................................................................................................................... **Output:**

{}

{}

...............................................................................................................................

Program No: 02

**Name of Program:** Write a Python program to add and print elements in a dictionary.

**Code:**

my\_cars = {}

print(“My empty dict: ”, cars)

my\_cars.setdefault(‘Cars’[]).append(“BMW”)

print(“First item added: ”, my\_cars)

my\_cars.setdefault(‘Cars’[]).append(“Toyota”)

print(“Second item added: ”, my\_cars)

my\_cars.setdefault(‘Cars’[]).append(“Honda”)

print(“Third item added: ”, my\_cars)

............................................................................................................................... **Output:**

My empty dict: {}

First item added: {‘Cars’: [‘BMW’]}

Second item added: {‘Cars’: [‘BMW’, ‘Toyota’]}

Third item added: {‘Cars’: [‘BMW’, ‘Toyota’, ‘Honda’]}

...............................................................................................................................

**Carefulness:** Ensure input validation for numerical values, maintain correct syntax and indentation, and define and call functions appropriately. Verify mathematical operations and logical conditions for accuracy.

**Remark:** Both programs effectively use dictionary operation to handle conditions based on user input, providing clear prompts and correct outputs.

Course Teacher Head of the Dept. Principal