```
#Name: Tasmim Noshrat Jahan Charu
#ID: 0242220005101844
#Section:63M1
#OOP2_LabReport01
```

```
#1. Python program to print "Hello Python"
print("Hello Python")
```

```
Hello Python
```

```
#2. Python program to do arithmetical operа
a = 5
b = 3
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
```

```
Addition: 8
Subtraction: 2
Multiplication: 15
Division: 1.6666666666666667
```

```
#3. Python program to find the area of a triang
base = 10
height = 5
area = 0.5 * base * height
print("Area of the triangle:", area)
```

```
Area of the triangle: 25.0
```

```
#4. Python program to solve a quadratic equatio
import math
```

```
a = 1
b = -7
c = 10

discriminant = b**2 - 4*a*c
root1 = (-b + math.sqrt(discriminant)) / (2*a)
root2 = (-b - math.sqrt(discriminant)) / (2*a)

print("Roots are:", root1, root2)
```

```
Roots are: 5.0 2.0
```

```
#5. Python program to swap two variables
x = 10
y = 20

x, y = y, x

print("Swapped values: x =", x, ", y =", y)
```

```
Swapped values: x = 20 , y = 10
```

```
#6. Python program to generate a random number
import random

print(random.randint(1, 100))
```

```
24
```

```
#7. Python program to convert kilometers to mil
kilometers = 5
```

```
conversion_factor = 0.621371
miles = kilometers * conversion_factor
print("Miles:", miles)
```

```
Miles: 3.106855
```

```
#8. Python program to convert Celsius to Fahren
celsius = 25
fahrenheit = (celsius * 9/5) + 32
print("Fahrenheit:", fahrenheit)
```

```
Fahrenheit: 77.0
```

```
#9. Python program to check if a number is posi
num = 3

if num > 0:
    print("Positive")
elif num == 0:
    print("Zero")
else:
    print("Negative")
```

```
Positive
```

```
#10. Python program to check if a number is odd
num = 7

if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

```
Odd
```

```
#11. Python program to check leap year
year = 2024

if (year % 4 == 0 and year % 100 != 0) or (year
    print("Leap year")
else:
    print("Not a leap year")
```

```
Leap year
```

```
#12. Python program to find the largest among t
a = 10
b = 15
c = 12

if a >= b and a >= c:
    largest = a
elif b >= a and b >= c:
    largest = b
else:
    largest = c

print("Largest number:", largest)
```

```
Largest number: 15
```

```
#13. Python program to check prime numbers
num = 29
```

```
if num > 1:
    for i in range(2, int(num/2)+1):
        if num % i == 0:
            print(num, "is not a prime number")
            break
    else:
        print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

```
29 is a prime number
```

```
#14. Python program to print all prime numbers
start = 10
end = 50

for num in range(start, end + 1):
    if num > 1:
        for i in range(2, int(num / 2) + 1):
            if num % i == 0:
                break
        else:
            print(num)
```

```
11
13
17
19
23
29
31
37
41
43
47
```

```
#15. Python program to find the factorial of a
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

num = 5
print("Factorial:", factorial(num))
```

```
Factorial: 120
```

```
#16. Python program to display the Fibonacci se
def fibonacci(n):
    a, b = 0, 1
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b

fibonacci(10)
```

```
0 1 1 2 3 5 8 13 21 34
```

```
#17. Python program to check Armstrong number
num = 153
order = len(str(num))
sum = 0
temp = num

while temp > 0:
    digit = temp % 10
```

```
    sum += digit ** order
    temp //= 10

if num == sum:
    print(num, "is an Armstrong number")
else:
    print(num, "is not an Armstrong number")
```

```
153 is an Armstrong number
```

```
#18. Python program to find Armstrong numbers i
start = 100
end = 1000

for num in range(start, end + 1):
    order = len(str(num))
    sum = 0
    temp = num

    while temp > 0:
        digit = temp % 10
        sum += digit ** order
        temp //= 10

    if num == sum:
        print(num)
```

```
153
370
371
407
```

```
#19. Python program to find the sum of natu
```

```
n = 10
sum = n * (n + 1) // 2
print("Sum of first", n, "natural numbers:'
```

```
Sum of first 10 natural numbers: 55
```

```
#20. Python program to find the factorial of a
def factorial(n):
    if n == 1:
        return 1
    else:
        return n * factorial(n-1)

num = 5
print("Factorial:", factorial(num))
```

```
Factorial: 120
```

```
#21. Python program to display the multiplicati
num = 7

for i in range(1, 11):
    print(num, 'x', i, '=', num * i)
```

```
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

```
#22. Python program to print the Fibonacci sequ
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

n_terms = 10
for i in range(n_terms):
    print(fibonacci(i), end=" ")
```

```
0 1 1 2 3 5 8 13 21 34
```

```
#23. Python program to find the sum of natural
def sum_of_natural_numbers(n):
    if n == 1:
        return 1
    else:
        return n + sum_of_natural_numbers(n-1)

n = 10
print("Sum of first", n, "natural numbers:", su
```

```
Sum of first 10 natural numbers: 55
```

```
#24. Python program to find the factors of a nu
num = 36

print("Factors of", num, "are:")
for i in range(1, num + 1):
```

```
    if num % i == 0:
        print(i)
```

```
Factors of 36 are:
1
2
3
4
6
9
12
18
36
```

```
#25. Python program to convert decimal to binar
dec = 344

print("Binary:", bin(dec))
print("Octal:", oct(dec))
print("Hexadecimal:", hex(dec))
```

```
Binary: 0b101011000
Octal: 0o530
Hexadecimal: 0x158
```

```
#26. Python program to find ASCII value of a ch
char = 'A'
print("The ASCII value of", char, "is", ord(cha
```

```
The ASCII value of A is 65
```

```
#27. Python program to find LCM of two numbers
def lcm(x, y):
    if x > y:
        greater = x
    else:
```

```
        greater = y

    while(True):
        if greater % x == 0 and greater % y ==
            lcm = greater
            break
        greater += 1

    return lcm

num1 = 54
num2 = 24
print("LCM:", lcm(num1, num2))
```

```
LCM: 216
```

```
#28. Python program to find HCF of two numb
def hcf(x, y):
    while(y):
        x, y = y, x % y
    return x

num1 = 54
num2 = 24
print("HCF:", hcf(num1, num2))
```

```
HCF: 6
```

```
#29. Python program to convert decimal to binar
def decimal_to_binary(n):
    if n > 1:
```

```
        decimal_to_binary(n // 2)
    print(n % 2, end='')

num = 10
decimal_to_binary(num)
```

```
1010
```

```
#30. Python program to find numbers divisible b
my_list = [12, 65, 54, 39, 102, 339, 221, 50, 7
divisor = 13

result = list(filter(lambda x: (x % divisor ==
print("Numbers divisible by", divisor, "are", r
```

```
Numbers divisible by 13 are [65, 39, 221]
```

```
#31. Python program to check if a string is a p
def is_palindrome(s):
    return s == s[::-1]

s = "madam"
if is_palindrome(s):
    print(s, "is a palindrome")
else:
    print(s, "is not a palindrome")
```

```
madam is a palindrome
```

```
#32. Python program to remove punctuation from
import string
```

```
s = "Hello!!!, he said --- and went."
no_punctuation = s.translate(str.maketrans('',
print(no_punctuation)
```

```
Hello he said  and went
```

```
#33. Python program to sort words in alphabetic
my_str = "Hello this is an example string"

words = my_str.split()
words.sort()

print("Sorted words:")
for word in words:
    print(word)
```

```
Sorted words:
Hello
an
example
is
string
this
```

```
#34. Python program to illustrate list oper
my_list = [1, 2, 3, 4, 5]

# Append an item
my_list.append(6)
print("After append:", my_list)

# Remove an item
my_list.remove(3)
print("After removal:", my_list)
```

```
print("After removal:", my_list)

# Reverse the list
my_list.reverse()
print("After reversing:", my_list)

# Sort the list
my_list.sort()
print("After sorting:", my_list)
```

```
After append: [1, 2, 3, 4, 5, 6]
After removal: [1, 2, 4, 5, 6]
After reversing: [6, 5, 4, 2, 1]
After sorting: [1, 2, 4, 5, 6]
```

```
#35. Python program to illustrate tuple ope
my_tuple = (1, 2, 3, 4, 5)

# Access elements
print("First element:", my_tuple[0])

# Slicing
print("Elements from index 1 to 3:", my_tup

# Find the length of the tuple
print("Length of the tuple:", len(my_tuple)
```

```
First element: 1
Elements from index 1 to 3: (2, 3, 4)
Length of the tuple: 5
```

```
#36. Python program to illustrate dictionar
my_dict = {'name': 'John', 'age': 25, 'city

# Access value by key
```

```
# Access value by key
print("Name:", my_dict['name'])

# Add new key-value pair
my_dict['email'] = 'john@example.com'
print("After adding email:", my_dict)

# Delete a key-value pair
del my_dict['age']
print("After deleting age:", my_dict)
```

```
Name: John
After adding email: {'name': 'John', 'age': 25, 'city': 'New York', 'email': 'john@e
After deleting age: {'name': 'John', 'city': 'New York', 'email': 'john@example.com'
```

```
#37. Python program to illustrate set operation
my_set = {1, 2, 3, 4, 5}

# Add an element
my_set.add(6)
print("After adding 6:", my_set)

# Remove an element
my_set.remove(3)
print("After removing 3:", my_set)

# Check for membership
print(4 in my_set)
```

```
After adding 6: {1, 2, 3, 4, 5, 6}
After removing 3: {1, 2, 4, 5, 6}
True
```

```
#38. Python program to find the power of a numb
def power(base, exp):
    if exp == 0:
        return 1
    else:
        return base * power(base, exp-1)

base = 2
exp = 3
print("Power:", power(base, exp))
```

```
Power: 8
```

```
#39. Python program to flatten a list
import itertools

list_of_lists = [[1, 2, 3], [4, 5], [6, 7, 8]]
flattened_list = list(itertools.chain(*list_of_
print(flattened_list)
```

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

```
#40. Python program to remove duplicates from a
my_list = [1, 2, 2, 3, 4, 4, 5]
unique_list = list(set(my_list))
print("Unique list:", unique_list)
```

```
Unique list: [1, 2, 3, 4, 5]
```

```
#41. Python program to check if a string is a s
main_str = "hello world"
sub_str = "world"
```

```
if sub_str in main_str:
    print(sub_str, "is a substring of", main_st
else:
    print(sub_str, "is not a substring of", mai
```

```
world is a substring of hello world
```

```
#42. Python program to calculate the sum of a l
my_list = [10, 20, 30, 40, 50]

total = sum(my_list)
print("Sum of the list:", total)

#43. Python program to count the occurrences of
my_list = [1, 2, 3, 2, 4, 2, 5]
count = my_list.count(2)
print("Number of occurrences of 2:", count)
```

```
Number of occurrences of 2: 3
```

```
#44. Python program to find the maximum and min
my_list = [10, 20, 5, 40, 50]

max_value = max(my_list)
min_value = min(my_list)

print("Maximum value:", max_value)
print("Minimum value:", min_value)
```

```
Maximum value: 50
Minimum value: 5
```

```
#45. Python program to reverse a string
my_str = "hello"
reversed_str = my_str[::-1]
print("Reversed string:", reversed_str)
```

```
Reversed string: olleh
```

```
#46. Python program to merge two dictionaries
dict1 = {'a': 1, 'b': 2}
dict2 = {'c': 3, 'd': 4}

merged_dict = {**dict1, **dict2}
print("Merged dictionary:", merged_dict)
```

```
Merged dictionary: {'a': 1, 'b': 2, 'c': 3, 'd': 4}
```

```
#47. Python program to convert a list of tuples
list_of_tuples = [('a', 1), ('b', 2), ('c', 3)]
dict_from_tuples = dict(list_of_tuples)
print("Dictionary:", dict_from_tuples)
```

```
Dictionary: {'a': 1, 'b': 2, 'c': 3}
```

```
#48. Python program to create a list of squares
numbers = [1, 2, 3, 4, 5]
squares = [x**2 for x in numbers]
print("List of squares:", squares)
```

```
List of squares: [1, 4, 9, 16, 25]
```

```
#49. Python program to find the common elements
list1 = [1, 2, 3, 4]
list2 = [3, 4, 5, 6]
```

```
common_elements = list(set(list1) & set(list2))
print("Common elements:", common_elements)
```

```
Common elements: [3, 4]
```

```
#50. Python program to check if all elements in
my_list = [1, 2, 3, 4, 5]

if len(my_list) == len(set(my_list)):
    print("All elements are unique")
else:
    print("There are duplicate elements")
```

```
All elements are unique
```