SMDYPSS's

DR. D. Y. PATIL CENTRE FOR MANAGEMENT & RESEARCH,

Chikhali, Pune - 412114.

Subject : Python Programming

Roll No:____________ **Division:______**

Name of Student:________________________

Assignment No-1

1. Print odd numbers between 1 to 100 using for loop

```
for i in range(1,100,2): #range(start_index, till_end, skip 2)
    print(i)

Output:
1
3
5
7
9
:
:
91
93
95
97
99
```

2. Python program to find out factorial of a given number

```
a=int(input("Enter the number"))
sum=1
for i in range(a,1,-1):
    sum=sum*i

print("Factorial of ",a ," is :",sum)

Output:
Enter the number5
Factorial of  5  is : 120
```

3. Write a python program to implement simple interest

```
print("Program for simple interest")

P=int(input("Enter the Principal Amount: "))
R=int(input("Enter the Rate: "))
T=int(input("Enter the Time: "))

SI = (P*R*T)/100; # Simple Interest calculation.

print("Simple Interest is :",SI) #prints Simple Interest.

Output:
Program for simple interest
Enter the Principal Amount: 5000
Enter the Rate: 15
Enter the Time: 1
Simple Interest is : 750.0
```

4. Write a python program to check whether the given string is palindrom or not using function

```
def palindrome(s):
    str = ""
    for i in s:
        str = i + str
    print("Reverse string: ",str)
    if s==str:
        print(s," is Palindrome")
    else:
        print(s," is not Palindrome")
print("To check given string is palindrom or not using function")
s=input("Enter the string: ")
palindrome(s)
```

Output:

```
To check given string is palindrom or not using function
Enter the string: JaaJ
Reverse string:  JaaJ
JaaJ  is Palindrome
```

5. Python program to check Armstrong number

```
# Python program to check if the number is an Armstrong number or not

num = int(input("Enter a number: "))

sum = 0

# find the sum of the cube of each digit
temp = num
while temp > 0:
   digit = temp % 10
   sum += digit ** 3
   temp //= 10

# display the result
if num == sum:
   print(num,"is an Armstrong number")
else:
   print(num,"is not an Armstrong number")


Output:
Enter a number: 153
153 is an Armstrong number
```

6. Python program to check smallest no between 3 numbers entered by user

```
print("Python program to check smallest no between 3 numbers entered by user ")
a = int(input("Enter the first no:"))
b = int(input("Enter the second no:"))
c = int(input("Enter the third no:"))

smallest = 0

if a < b and a < c :
    smallest = a
if b < a and b < c :
    smallest = b
if c < a and c < b :
    smallest = c

print(smallest, "is the smallest of three numbers.")
```

Output:

```
Python program to check smallest no between 3 numbers entered by user
Enter the first no:12
Enter the second no:67
Enter the third no:34
12 is the smallest of three numbers.
```

7. Python program to find out nth Fibonacci series

```
# Program to display the Fibonacci sequence up to n-th term

nterms = int(input("How many terms? "))

n1, n2 = 0, 1
count = 0

print("Fibonacci sequence:")
while count < nterms:
    print(n1)
    nth = n1 + n2
    # update values
    n1 = n2
    n2 = nth
    count += 1
```

Output:

```
How many terms? 5
Fibonacci sequence:
0
1
1
2
3
```

SMDYPSS's

DR. D. Y. PATIL CENTRE FOR MANAGEMENT & RESEARCH,

Chikhali, Pune - 412114.

Subject : Python Programming

Roll No:____________ **Division:______**

Name of Student:________________________

Assignment-2

1. Python program to find the union of 2 sets

```
# Find the union of 2 sets

A = {2, 4, 5, 6}  # set-1
B = {4, 6, 7, 8}  # set-2

print("Set A:",A)
print("Set B:",B)
print("A U B:", A.union(B))   # union()



Output:
Set A: {2, 4, 5, 6}
Set B: {8, 4, 6, 7}
A U B: {2, 4, 5, 6, 7, 8}
```

2. Python program to merge two list into 1 list in sorted format input: A=[45,78, 92] and B=[30,42,85]

```
#Python program to merge two list into 1 list in sorted format input:

A=[45,78,92]
B=[30,42,85]

print("List A:",A)
print("List B:",B)

C=A+B # merge 2 list
C.sort() #sort the list

print("Sorted merged list: ",C)

Output:
List A: [45, 78, 92]
List B: [30, 42, 85]
Sorted merged list:  [30, 42, 45, 78, 85, 92]
```

3. Write a program to implement any five string functions with proper example

```
#string
a = "Hello, World!"
print(a[1])

print(len(a))

for x in "banana":
    print(x)

txt = "The best things in life are free!"
print("free" in txt)


if "free" in txt:
    print("Yes, 'free' is present.")


print("expensive" not in txt)
if "expensive" not in txt:
    print("No, 'expensive' is NOT present.")

b = "Hello, World!"
print(b[2:5])

#Get the characters from the start to position 5 (not included):

b = "Hello, World!"
print(b[:5])

#Get the characters from position 2, and all the way to the end:

b = "Hello, World!"
print(b[2:])

'''
Negative index :
Last character start with -1
Second last character -2
'''
b = "Hello, World!"
print(b[-5:-2]) #orl

a = "Hello, World!"
```

```
print(a.upper())

a = "Hello, World!"
print(a.lower())

a = "   Hello, World!   " #remove white spaces
print(a.strip()) # returns "Hello, World!"

Output:
e
13
b
a
n
a
n
a
True
Yes, 'free' is present.
True
No, 'expensive' is NOT present.
llo
Hello
llo, World!
orl
HELLO, WORLD!
hello, world!
Hello, World!
```

4. Write a program to implement tuple and implement its functions

```
#Python program to implement tuple:

#tuple has 2 functions count and index
t=(10,20,30,10,10)
print(t)
print("t.count(10): ",t.count(10))
print("t.index(30): ",t.index(30))



t1= (50,20,10,40,30)
print("t1: ",t1)
print("t1[0]: ",t1[0])

tuple1=("Disco", 4.5, 10)
print(tuple1)

tuple2=tuple1+ ("hello",100)
print(tuple2)

print(tuple2[0:2])
print(tuple2[3:5])
print(len(tuple2))

sorted_t=sorted(t)
print("sorted_t: ",sorted_t)
```

Output:

```
(10, 20, 30, 10, 10)
t.count(10):  3
t.index(30):  2
t1:  (50, 20, 10, 40, 30)
t1[0]:  50
('Disco', 4.5, 10)
('Disco', 4.5, 10, 'hello', 100)
('Disco', 4.5)
('hello', 100)
5
sorted_t:  [10, 10, 10, 20, 30]
```

5. Write a program to implement set and implement its functions

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

# Adding 5 into the set
my_set.add(5)
print("After adding 5 into the set: ",my_set)

# Checking element present in the set or not
print("Is 3 present in the set: ", 3 in my_set)
print("Is 6 present in the set: ", 6 in my_set)

# Removing the element in the set
my_set.remove(4)
print("Set after removing 4: ",my_set)

# calculating the length of the set
print("Length of the set: ",len(my_set))

set1 = {1,2,3,4,5}
set2 = {4,5,6,7,8}

# Union of two sets
print("Union of two sets: ",set1.union(set2))

# Intersection of two sets
print("Intersection of two sets: ",set1.intersection(set2))

# Difference of two sets
print("Difference of two sets: ",set1.difference(set2))

# Symmetric difference of two sets
print("Symmetric difference of two sets: ",set1.symmetric_difference(set2))
```

Output:

```
After adding 5 into the set:  {1, 2, 3, 4, 5}
Is 3 present in the set:  True
Is 6 present in the set:  False
Set after removing 4:  {1, 2, 3, 5}
Length of the set:  4
Union of two sets:  {1, 2, 3, 4, 5, 6, 7, 8}
Intersection of two sets:  {4, 5}
Difference of two sets:  {1, 2, 3}
Symmetric difference of two sets:  {1, 2, 3, 6, 7, 8}
```

6. Write a python program for the following.

   Create list of fruits

   Add new fruit in list.

   sort the list.

   delete last fruit name from list.

```
fruits = ["Banana", "Pineapple", "Apple"]

# Add new fruit into the list
fruits.append("mango")
print(fruits)

# Sort the list
fruits.sort()
print("The sorted list: ",fruits)

# Removes the last element
del fruits[-1]
print("After removing last element from the list:", fruits)
```

Output:

```
['Banana', 'Pineapple', 'Apple', 'mango']
The sorted list:  ['Apple', 'Banana', 'Pineapple', 'mango']
After removing last element from the list: ['Apple', 'Banana', 'Pineapple']
```

7. Python program to implement Dictionary and its 5 functions

```
# creating the dictionary
my_dict = {'apple':3, 'orange':6, 'banana':2}

# Accessing the value by key
print("The value of apple: ", my_dict['apple'])

# Adding a new key-value pair
my_dict['grape'] = 4
print("After adding 'grape': ", my_dict)

# Removing key-vaue pair
del my_dict['banana']
print("After removing 'banana: ", my_dict)

# Checking if a key exists
if 'apple' in my_dict:
    print("The key 'apple' exists in the dictionary.")
else:
    print("The key 'apple' does not exist in the dictionary.")

# getting the number key-value pairs
num = len(my_dict)
print("The length of the dictionary: ",num)
```

Output:

```
The value of apple:  3
After adding 'grape':  {'apple': 3, 'orange': 6, 'banana': 2, 'grape': 4}
After removing 'banana:  {'apple': 3, 'orange': 6, 'grape': 4}
The key 'apple' exists in the dictionary.
The length of the dictionary:  3
```

SMDYPSS's

DR. D. Y. PATIL CENTRE FOR MANAGEMENT & RESEARCH,

Chikhali, Pune - 412114.

Subject : Python Programming

Roll No:____________ **Division:______**

Name of Student:________________________

Assignment-3

1. Python program to implement following using functions 1. Area of circle 2. Area of rectangle 3. Area of square 4. Area of triangle

```
import math

def area_of_circle(radius):
   print ("Area of circle: ",math.pi * radius ** 2)

def area_of_rectangle(length, width):
   print ("Area of rectangle: ",length*width)

def area_of_square(side):
   print ("Area of square:" ,side**2)

def area_of_triangle(base, height):
   print ("Area of triangle: ",(0.5) * base * height)

radius = float(input("Enter the radius: "))
area_of_circle (radius)

length = float(input("Enter the length: "))
width = float(input("Enter the width: "))
area_of_rectangle (length, width)

side = float(input("Enter the side: "))
area_of_square (side)

base = float(input("Enter the base: "))
height = float(input("Enter the height: "))
area_of_triangle (base, height)
```

Output:

```
Enter the radius: 5
Area of circle:  78.53981633974483
Enter the length: 5
Enter the width: 4
Area of rectangle:  20.0
Enter the side: 5
Area of square: 25.0
Enter the base: 10
Enter the height: 5
Area of triangle:  25.0
```

2. Write a program to implement lambda function in python?

```
add = lambda x,y: x+y

x = int(input("Enter the value of x: "))
y = int(input("Enter the value of y: "))

result = add(x,y)
print(f"The addition of {x} and {y} is: {result}")

Output:
Enter the value of x: 5
Enter the value of y: 10
The addition of 5 and 10 is: 15
```

3. **(multilevel inheritance)**

   class Person with name, age

   Member function: constructor(), Display()

   Class Student with roll_no, branch

   Member function: constructor(), Display()

   Class Exam with subject[5], total, percentage

   Member function: constructor(), Display()

   Class Student is inherited by class Person & class Exam is inherited by class Student

   Write a program to read all information & display it. **(use constructor)**

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def display(self):
        print("Name: ", self.name, ", Age: ", self.age)

class Student(Person):
    def __init__(self, name, age, rollno, branch):
        super().__init__(name, age)
        self.rollno = rollno
        self.branch = branch

    def display(self):
        super().display()
        print("Rollno: ", self.rollno, ", Branch: ", self.branch)

class Exam(Student):
    def __init__(self, name, age, rollno, branch,m1,m2,m3,m4,m5):
        super().__init__(name, age, rollno, branch)
        self.m1 = m1
        self.m2 = m2
        self.m3 = m3
        self.m4 = m4
        self.m5 = m5
        self.total = self.m1+self.m2+self.m3+self.m4+self.m5
        self.percentage = self.total/5

    def display(self):
        super().display()
        print("Marks: ", self.total)
        print("Percentage: ", self.percentage)

name = input("Enter name: ")
age = int(input("Enter age: "))
rollno = int(input("Enter RollNo: "))
branch = input("Enter Branch: ")
m1 = int(input("Enter Marks: "))
m2 = int(input("Enter Marks: "))
m3 = int(input("Enter Marks: "))
m4 = int(input("Enter Marks: "))
m5 = int(input("Enter Marks: "))

e1 = Exam(name, age, rollno, branch, m1, m2, m3, m4, m5)
e1.display()
```

Output:

```
Enter name: xyz
Enter age: 21
Enter RollNo: 122159
Enter Branch: commerce
Enter Marks: 95
Enter Marks: 96
Enter Marks: 94
Enter Marks: 89
Enter Marks: 96
Name:  xyz , Age:  21
Rollno:  122159 , Branch:  commerce
Marks:  470
Percentage:  94.0
```

4. **(multiple inheritance)**

   class InternalExam with subject[5] marks out of 20 i.e. list, total

   Member function: Constructor for read data(), disp(), Calculate ()

   class ExternalExam with subject[5] marks out of 80i.e. list, total

   Member function: Constructor for read data(), disp(), Calculate ()

   class Result_Combine with with subject[5] marks out of 100 (internal + external) , total of marks, percentage

   Member function: Constructor for read data(), disp(), Calculate ()

   Class Result_Combine is inherited by class InternalExam & ExternalExam

   Write a program to read all information & display it. **(use super() keyword)**

```
class InternalExam:
    def __init__(self):
        self.marks = []
        self.total = 0

    def read_data(self):
        for i in range(5):
            mark = int(input(f"Enter mark out of 20: "))
            self.marks.append(mark)

    def disp(self):
        print("Internal Exam:")
        for i in range(5):
            print(f" {self.marks[i]}/20")
        print(f"Total marks: {self.total}")

    def calculate(self):
        self.total = sum(self.marks)


class ExternalExam:
    def __init__(self):
        self.marks = []
        self.total = 0

    def read_data1(self):
        for i in range(5):
            mark = int(input(f"Enter mark out of 80: "))
            self.marks.append(mark)

    def disp1(self):
        print("External Exam:")
        for i in range(5):
            print(f" {self.marks[i]}/80")
        print(f"Total marks: {self.total}")

    def calculate(self):
        self.total = sum(self.marks)


class Result_Combine(InternalExam, ExternalExam):
    def __init__(self):
        super().__init__()

    def disp(self):
        print(f"Total marks: {self.total}")
        print(f"Percentage: {self.total / 500 * 100}%")

    def calculate(self):
        super().calculate()
        self.total += sum(self.marks)
        super().calculate()

rc = Result_Combine()
rc.read_data()
rc.read_data1()
rc.calculate()
rc.disp()
```

Output:

```
Enter mark out of 20: 19
Enter mark out of 20: 18
Enter mark out of 20: 16
Enter mark out of 20: 19
Enter mark out of 20: 19
Enter mark out of 80: 75
Enter mark out of 80: 78
Enter mark out of 80: 76
Enter mark out of 80: 75
Enter mark out of 80: 74
Total marks: 469
Percentage: 93.8%
```

5. Write a program to implement delegation and container function in python?

```
class Container:
    def __init__(self):
        self.items = []

    def add_items(self,item):
        self.items.append(item)

    def remove_items(self,item):
        self.items.remove(item)

    def print_items(self):
        for item in self.items:
            print(item)

class Delegator:
    def __init__(self):
        self.container = Container()

    def add_item(self, item):
        self.container.add_items(item)

    def remove_item(self, item):
        self.container.remove_items(item)

    def print_items(self):
        self.container.print_items()


Delegator = Delegator()
Delegator.add_item("Item 1")
Delegator.add_item("Item 2")
Delegator.add_item("Item 3")
Delegator.print_items()
Delegator.remove_item("Item 2")
print("\n After remove an 'Item2'")
Delegator.print_items()
```

```
Output:
Item 1
Item 2
Item 3

 After remove an 'Item2'
Item 1
Item 3
```

SMDYPSS's

DR. D. Y. PATIL CENTRE FOR MANAGEMENT & RESEARCH,

Chikhali, Pune - 412114.

Subject : Python Programming

Roll No:____________ **Division:______**

Name of Student:________________________

Assignment-4

1. Write a python program to extract e-mail address from file

```
import re

def extract_email_address(file_path):
    with open(file_path, 'r') as file:
        data = file.read()
        email_regex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b'
        emails = re.findall(email_regex,data,re.IGNORECASE)
        return emails

file_path = "emails.txt"
emails = extract_email_address(file_path)
for email in emails:
    print(email)

Output:
xyz2002@gmail.com
abc123@gmail.com
```

2. Write a python program to check password validation using regular expression.

```
import re

def validate_password(password):
    if len(password)<8:
        return False
    if not re.search(r'[A-Z]',password):
        return False
    if not re.search(r'[a-z]',password):
        return False
    if not re.search(r'\d',password):
        return False
    if not re.search(r'[\W_]',password):
        return False

    return True

password = input("Enter a password: ")
if validate_password(password):
    print("valid Password")
else:
    print("Invalid Password")

Output:
Enter a password: Hello@123
valid Password
```

3. Write a python program to extract e-mail address from web page

```
import re
import requests

def extract_emails(url):
    response = requests.get(url)
    content = response.text

    email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
    emails = re.findall(email_pattern, content, re.IGNORECASE)

    return emails

url = 'https://en.wikipedia.org/wiki/Email_address'
emails = extract_emails(url)

if emails:
    print("Email addresses found:")
    for email in emails:
        print(email)
else:
    print("No email addresses found.")

Output:
Email addresses found:
john.smith@example.com
jsmith@example.com
john.smith@example.org
John..Doe@example.com
```

4. Write a multithreading program, where one thread prints square of a number and another thread prints cube of a number. Also display the total time taken for the execution.

```
import threading
import time

def calculate_square(number):
    print(f"Square: {number*number}")

def calculate_cube(number):
    print(f"Cube: {number*number*number}")

number = 5

square_thread = threading.Thread(target=calculate_square(number))
cube_thread = threading.Thread(target=calculate_cube(number))

start_time = time.time()
square_thread.start()
cube_thread.start()
square_thread.join()
cube_thread.join()
print(f"Time taken: {time.time() - start_time}")

Output:
Square: 25
Cube: 125
Time taken: 0.0009264945983886719
```

5. Write a multithreading program, where one thread prints “Happy Birthday” and another thread prints “It’s a time for party”. Also display the total time taken for the execution.

```
import threading
import time

def print_wish():
   print("Happy Birthday!")

def print_message():
   print("It's time for party")

wish_thread = threading.Thread(target=print_wish())
message_thread = threading.Thread(target=print_message())

start_time = time.time()
wish_thread.start()
message_thread.start()
wish_thread.join()
message_thread.join()
print(f"Time taken: {time.time() - start_time}")

Output:
Happy Birthday!
It's time for party
Time taken: 0.0008869171142578125
```

SMDYPSS's

DR. D. Y. PATIL CENTRE FOR MANAGEMENT & RESEARCH,

Chikhali, Pune - 412114.

Subject : Python Programming

Roll No:____________ **Division:______**

Name of Student:________________________

Assignment-5

1. Write a python program to copy the content of 1 file to another

```
with open("write1.txt",'r') as fr:
    with open("read1.txt",'w') as fw:
        for l in fr:
            fw.write(l)
fr.close()
fw.close()
```

Output:

```
Write1.txt:
Hello World!

Read1.txt:
Hello World!
```

2. Write a program to store 1 to 20 numbers into the file and display only even nos from file

```
def store_numbers():
    with open("numbers.txt", "w") as file:
        for number in range(1, 21):
            file.write(str(number) + "\n")

def display_even_numbers():
    with open("numbers.txt", "r") as file:
        print("Even numbers from the file:")
        for line in file:
            number = int(line.strip())
            if number % 2 == 0:
                print(number)

# Store numbers 1 to 20 into the file
store_numbers()

# Display only the even numbers from the file
display_even_numbers()
```

Output:

```
Numbers.txt:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Even Numbers:
```

```
2 4 6 8 10 12 14 16 18 20
```

3. Write a python program to create “product” collection with fields” (ID, name, purchase_price, sale_price and quantity) in mongoDB. Perform the following operations.

    Display product details of product name = “Pen” department

    Delete product with ID - 2103

    Update quantity with new quantity for product ID -2102

    Display all product details from product collection

```
import pymongo

if __name__ == "__main__":
    print("welcome to pymongo")
    # Creates Mongo Cient
    client = pymongo.MongoClient("mongodb://localhost:27017/")
    print(client)

    # creates database
    db = client['product']

    # creates collection
    collection = db['product']

    # insert many items into the collection
    insertThese = [
        {
            '_id':2101,
            'name':'sharpner',
            'purchase_price':5,
            'sale_price':8,
            'quantity':30
        },
        {
            '_id':2102,
            'name':'eraser',
            'purchase_price':2,
            'sale_price':5,
            'quantity':40
        },
        {
            '_id':2103,
            'name':'pen',
            'purchase_price':5,
            'sale_price':8,
            'quantity':60
        },
        {
            '_id':2104,
            'name':'pencil',
            'purchase_price':3,
            'sale_price':6,
            'quantity':30
        },
    ]

    collection.insert_many(insertThese)

    # display product details of product name = 'pen'
    Pname = collection.find({'name':'pen'})
    for item in Pname:
        print(item)

    # delete the product with ID-2103
    collection.delete_one({'_id':2103})
    print("The product with ID-52103 has been deleted")

    # update quantity with new quantity for product Id-2102
    collection.update_one({'_id':2102},{'$set':{'quantity':60}})
    print("The quantity of product Id-2102 has been updated")

    # display all the details from the product collection
    Pname = collection.find()
    for item in Pname:
        print(item)
```

Output:

```
welcome to pymongo

MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True)

{'_id': 2103, 'name': 'pen', 'purchase_price': 5, 'sale_price': 8, 'quantity': 60}

The product with ID-2103 has been deleted
The quantity of product Id-2102 has been updated

{'_id': 2101, 'name': 'sharpner', 'purchase_price': 5, 'sale_price': 8, 'quantity': 30}
{'_id': 2102, 'name': 'eraser', 'purchase_price': 2, 'sale_price': 5, 'quantity': 60}
{'_id': 2104, 'name': 'pencil', 'purchase_price': 3, 'sale_price': 6, 'quantity': 30}
```

4. Write a python program to create “Employee” collection with fields” (ID, name, phone_no, email_id and department) in mongoDB. Perform the following operations.

    Display employee details of name = “John Nair” department

    Delete employee with ID - 103

    Update phone_no with new phone_no of employee ID -2102

    Display all employee details from emloyee collection

```
import pymongo

if __name__ == "__main__":

    # creates mongo client
    client = pymongo.MongoClient("mongodb://localhost:27017/")
    print(client)

    # creates database
    db = client['Employee']

    # creates collection
    collection = db['Employee']

    # insert multiple items into the collection
    InsertThese = [
        {
            '_id':103,
            'name':"Healy kashyap",
            'phone_no':1234567890,
            'email_id':'Healy@gmail.com',
            'department':"production"
        },
        {
            '_id':101,
            'name':"John Nair",
            'phone_no':3987654321,
            'email_id':'John@gmail.com',
            'department':"packing"
        },
        {
            '_id':2102,
            'name':"Justin Biber",
            'phone_no':6532589750,
            'email_id':'Justin@gmail.com',
            'department':"painting"
        },
        {
            '_id':104,
            'name':"john",
            'phone_no':7452136010,
            'email_id':'john@gmail.com',
            'department':"production"
        },
    ]

    collection.insert_many(InsertThese)

    # display the employee details of name ="John Nair"
    Ename = collection.find_one({'name':"John Nair"})
    print(Ename)

    # delete the employee with Id- 103
    collection.delete_one({'_id':103})
    print("deleted the employee with id-103")

    # update the phone number with new phone number with id-2102
    collection.update_one({'_id':2102}, {'$set':{'phone_no':6542136010}})
    print("updated the phone number with id-2102")

    # display all the data into the colection
    allDocs = collection.find()
    for doc in allDocs:
        print(doc)
```

Output:

```
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False,
connect=True)
```

```
{'_id': 101, 'name': 'John Nair', 'phone_no': 3987654321, 'email_id':
'John@gmail.com', 'department': 'packing'}

deleted the employee with id-103

updated the phone number with id-2102

{'_id': 101, 'name': 'John Nair', 'phone_no': 3987654321, 'email_id':
'John@gmail.com', 'department': 'packing'}
{'_id': 2102, 'name': 'Justin Biber', 'phone_no': 6542136010, 'email_id':
'Justin@gmail.com', 'department': 'painting'}
{'_id': 104, 'name': 'john', 'phone_no': 7452136010, 'email_id':
'john@gmail.com', 'department': 'production'}
```