CO5

Q(1). Write a Python program to read a file line by line and store it into a list.

```
# Function to write content to a file
def write_to_file(file_path, content):
    with open(file_path, 'w') as file:
        file.write(content)

# Function to read a file line by line and store lines in a list
def read_file_to_list(file_path):
    lines = []
    with open(file_path, 'r') as file:
        for line in file:
            lines.append(line.strip())  # Remove trailing newline characters

    return lines

# Example usage
file_path = 'example.txt'

# Write content to the file
content_to_write = """Line 1: This is the first line.
Line 2: This is the second line.
Line 3: This is the third line."""
write_to_file(file_path, content_to_write)

# Read the file and store lines in a list
lines_read = read_file_to_list(file_path)

# Print the lines
```

```
print("Lines read from the file:")
for line in lines_read:
    print(line)
##
def copy_odd_lines(input_file_path, output_file_path):
    with open(input_file_path, 'r') as input_file:
        with open(output_file_path, 'w') as output_file:
            # Enumerate is used to get both the line and its index
            for index, line in enumerate(input_file):
                # Check if the line number is odd (using 0-based indexing)
                if index % 2 != 0:
                    output_file.write(line)


# Example usage
input_file_path = 'example.txt'
output_file_path = 'newexample.txt'


copy_odd_lines(input_file_path, output_file_path)


print(f"Odd lines copied from '{input_file_path}' to '{output_file_path}'.")
```

Q(2). Python program to copy odd lines of one file to other

```
# Function to write content to a file
def write_to_file(file_path, content):
    with open(file_path, 'w') as file:
        file.write(content)


# Function to read a file line by line and store lines in a list
def read_file_to_list(file_path):
    lines = []
    with open(file_path, 'r') as file:
        for line in file:
            lines.append(line.strip())  # Remove trailing
newline characters

    return lines

# Example usage
file_path = 'example.txt'

# Write content to the file
content_to_write = """Line 1: This is the first line.
Line 2: This is the second line.
Line 3: This is the third line."""
write_to_file(file_path, content_to_write)

# Read the file and store lines in a list
lines_read = read_file_to_list(file_path)

# Print the lines
print("Lines read from the file:")
for line in lines_read:
    print(line)
##
def copy_and_print_odd_lines(input_file_path,
output_file_path):
    odd_lines = []
    with open(input_file_path, 'r') as input_file:
        with open(output_file_path, 'w') as output_file:
            for index, line in enumerate(input_file):
                if index % 2 == 0:
                    output_file.write(line)
                    odd_lines.append(line.strip())

    print("Odd lines copied to '{}' and their contents:".format(output_file_path))
    for line_content in odd_lines:
        print(line_content)

# Example usage
input_file_path = 'example.txt'
output_file_path = 'newexample.txt'

copy_and_print_odd_lines(input_file_path, output_file_path)
```

Q(3). Write a Python program to read each row from a given csv file and print a list of strings.

```
import csv

def write_to_csv(filename, data):
    """
    Write data to a CSV file.
    :param filename: Name of the CSV file
    :param data: List of lists representing rows of data
    """
    with open(filename, 'w', newline='') as csvfile:
        csv_writer = csv.writer(csvfile)
        csv_writer.writerows(data)
    print(f'Data has been written to {filename}')

def read_csv_and_print(filename):
    """
    Read each row from a CSV file and print a list of strings.
    :param filename: Name of the CSV file
    """
    with open(filename, 'r') as csvfile:
        csv_reader = csv.reader(csvfile)
        for row in csv_reader:
            print(', '.join(row))

# Example usage:
# Writing to CSV
data_to_write = [
    ['Name', 'Age', 'City'],
    ['John Doe', '25', 'New York'],
    ['Jane Smith', '30', 'San Francisco'],
    ['Bob Johnson', '22', 'Los Angeles']
]
write_to_csv('example.csv', data_to_write)

# Reading from CSV and printing
print('\nReading from CSV and printing:')
read_csv_and_print('example.csv')
```

Q(4). Write a Python program to read specific columns of a given CSV file and print the content of the columns.

```
import csv
def create_csv(filename, data):
    """
    Create a CSV file and write data to it.
    :param filename: Name of the CSV file
    :param data: List of lists representing rows of data
    """
    with open(filename, 'w', newline='') as csvfile:
        csv_writer = csv.writer(csvfile)
        csv_writer.writerows(data)
    print(f'Data has been written to {filename}')

def read_specific_columns(filename, columns):
    """
    Read specific columns from a CSV file and print the content.
    :param filename: Name of the CSV file
    :param columns: List of column indices to be read and printed
    """
    with open(filename, 'r') as csvfile:
        csv_reader = csv.reader(csvfile)
        for row in csv_reader:
            selected_data = [row[i] for i in columns]
            print(', '.join(selected_data))

# Example usage:
# Creating CSV
data_to_write = [
    ['Name', 'Age', 'City'],
    ['John Doe', '25', 'New York'],
    ['Jane Smith', '30', 'San Francisco'],
    ['Bob Johnson', '22', 'Los Angeles']
]
create_csv('example.csv', data_to_write)

# Reading specific columns from CSV and printing
print('\nReading specific columns from CSV and printing:')
# Assuming the columns are 0 (Name) and 2 (City)
```

```
read_specific_columns('example.csv', [1, 2])
```

Q(5). Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

```
import csv


def write_dict_to_csv(filename, data_dict):
    """
    Write a Python dictionary to a CSV file.
    :param filename: Name of the CSV file
    :param data_dict: Python dictionary to be written to CSV
    """
    with open(filename, 'w', newline='') as csvfile:
        csv_writer = csv.DictWriter(csvfile,
fieldnames=data_dict.keys())
        csv_writer.writeheader()
        csv_writer.writerow(data_dict)
    print(f'Dictionary has been written to {filename}')


def read_csv_and_display(filename):
    """
    Read a CSV file and display its content.
    :param filename: Name of the CSV file
    """
    with open(filename, 'r') as csvfile:
        csv_reader = csv.DictReader(csvfile)
        for row in csv_reader:
            print(row)


# Example usage:
```

```
# Writing dictionary to CSV
data_to_write = {'Name': 'John Doe', 'Age': '25', 'City': 'New York'}
write_dict_to_csv('example.csv', data_to_write)

# Reading and displaying content from CSV
print('\nReading and displaying content from CSV:')
read_csv_and_display('example.csv')
```