

Python File Handling

1.

a. Write a Python program to read a file line by line and store it into a list.

```
# Open the file in write mode ('w') - this will create the file if it doesn't exist
file = open("file2.txt", "w")

# Write multiple lines of text into the file
file.write("hello world!\nwelcome to python programing\nThank you")

# Close the file after writing
file.close()

# Create an empty list to store the lines
list = []

# Open the same file in read mode ('r')
file = open("file2.txt", "r")

# Read each line from the file, remove the whitespace/tab/newline character, and store it in the list
list = [line.strip() for line in file]

# Close the file after reading
file.close()

# Print the list containing all lines from the file
print(list)
```

b. Python program to copy odd lines of one file to other

```
# Open file3.txt in write mode and write multiple lines to it
file = open("file3.txt", "w")
file.write("This is the firstline\nThis is the secondline\nThis is the thirdline\nThis is the fourthline")
file.close()

# Open the file in read mode and read the first three lines
file = open("file3.txt", "r")
print(file.readline()) # Reads the first line
print(file.readline()) # Reads the second line
print(file.readline()) # Reads the third line
file.close()

# Copy only odd-numbered lines from file3.txt to file4.txt
```

```

with open("file3.txt", "r") as file1:      # Open source file for reading
    with open("file4.txt", "w") as file2:  # Open destination file for writing
        line_number = 1                    # Initialize line counter
        for line in file1:                 # Iterate through each line in file1
            if line_number % 2 != 0:        # Check if the line number is odd
                file2.write(line)           # Write odd lines to file2
            line_number += 1                # Increment line counter

print("Odd lines written to file4.txt:")

# Open file4.txt and display all its lines using a loop
with open("file4.txt", "r") as file2:
    for line in file2:
        print(line.strip())               # Print each line without extra newline

```

2. Write a Python program to write a dictionary to a CSV file, then read the same CSV file to display its contents and print each row as a list of strings.

```

import csv

# Define the header (field names) for the CSV file
field_name = ['No', 'Company', 'Car Model']

# Create a list of dictionaries, each representing a row
car = [
    {'No': 1, 'Company': 'Ferrari', 'Car Model': 'GH'},
    {'No': 2, 'Company': 'BMW', 'Car Model': 'X5'},
    {'No': 3, 'Company': 'Maruti Suzuki', 'Car Model': 'Swift'},
    {'No': 4, 'Company': 'Audi', 'Car Model': 'RS7'},
    {'No': 5, 'Company': 'Toyota', 'Car Model': 'Fortuner'}
]

# ----- Writing to the CSV file -----
with open('car.csv', 'w', newline='') as csvfile:
    write = csv.DictWriter(csvfile, fieldnames=field_name)  # Create DictWriter
                                                            # object
    write.writeheader()                                     # Write header row
    write.writerows(car)                                    # Write multiple rows from list of dictionaries

```

```
# ----- Reading from the CSV file -----  
with open('car.csv', newline='') as csvfile:  
    d = csv.reader(csvfile)    # Create a CSV reader object  
    for r in d:                # Iterate through each row in the CSV file  
        print(','.join(r))     # Join each element in row with commas and print  
        print(r)               # Display the same row as a list of strings
```