CSC248 – Fundamentals of Data Structure
Academic Session October 2023 – February 2024
Lab Assignment 3 – Linked List (Built-In)

Course Outcomes (CO)	LO1	LO2	LO3
CO1			
CO2	√	√	√
CO3			

Given the following `Friend` and `LinkedList` ADT

```
public class Friend
{
	private int idno;
	private String name, hpno, email;

	public Friend (int id, String n, String hp, String em) {…}
	public int getId() {…}
	public String getName() {…}
	public String getHP() {…{
	public String getEmail() {…}
	public void setName(String n) {…}
	public void setHP(String hp) {…}
	public void setEmail(String mail) {…}

}

public class LinkedList
{
	public LinkedList();
	public int size();
	public void add(int index, Object element);
	public Object get(int index);
	public Object set(int index, Object element);
	public Object remove(int index)
}
```

Write a Java program to:

1. Write a complete program of `Friend` class.

```java
public class Friend {
    private int idno;
    private String name, hpno, email;

    public Friend(int idno, String name, String hpno, String email) {
        this.idno = idno;
        this.name = name;
        this.hpno = hpno;
        this.email = email;
    }

    public int getIdno() {
        return this.idno;
    }

    public void setIdno(int idno) {
        this.idno = idno;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getHpno() {
        return this.hpno;
    }

    public void setHpno(String hpno) {
        this.hpno = hpno;
    }

    public String getEmail() {
        return this.email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

2. Write an application program by implementing a **MENU SELECTION** to do the following tasks.

   a. Create a linked list named `sList` then insert some records (identify by the user) into `sList`. Verify each record to ensure that there is no duplicate record.

   b. Ask user to enter id no for viewing the record. If the record exists the view it on the screen otherwise display an appropriate message
   c. Remove the record form the `sList`. If the `slist` is empty or has only one record then, display an appropriate message.
   d. Update record that it's based on `idno` given by the user. If the record exists replace its value of `hpno` and `email` also given by the user, otherwise display an appropriate message.
   e. To print all record from the `sList`

name.txt

Ahmad;Aiman;Alif;Amir;Aziz;Bakar;Basir;Daud;Faisal;Faiz;Faris;Fikri;Ghani;Hafiz;Hamid;Hasan;Hisham;Iqbal;Ismail;Izzat;Jalil;Jamal;Johan;Kamal;Karim;Khairi;Latif;Mahmud;Mansor;Nasir;Omar;Osman;Rahim;Rahman;Rais;Rashid;Ridzuan;Rosli;Saad;Sabri;Safwan;Sahar;Saleh;Salim;Samad;Shafiq;Sharif;Syafiq;Tahir;Umar;Usman;Wahab;Yahya;Yusof;Zahari;Zain;Zaki;Zamri;Zulkifli;Aida;Aina;Aisyah;Alia;Amira;Azura;Balqis;Batrisyia;Diana;Elyana;Farah;Fatin;Hafizah;Hamidah;Haslinda;Husna;Intan;Izzah;Jasmin;Julia;Kamalia;Karina;Khadijah;Latifah;Liyana;Mahirah;Maisarah;Nadia;Najwa;Natasha;Nurul;Puteri;Qistina;Rahimah;Raihanah;Rania;Rashidah;Ridzuanah;Rosnah;Saadiah;Sabrina;Safiah;Sakinah;Salimah;Sara;Shafiqah;Sharifah;Syafiqah;Tahirah;Umi;Wahidah;Yasmin;Yusnita;Zaharah;Zainab;Zakiyah;Zamzuriah;Zulaikha

Abdullah;Abu;Ahmad;Ali;Amin;Aziz;Bakar;Basir;Daud;Faisal;Ghani;Hamid;Hasan;Hussein;Ibrahim;Ismail;Jalil;Jamal;Johan;Kamal;Karim;Kassim;Latif;Mahmud;Mansor;Mohamad;Mohamed;Mohd;Muhammad;Musa;Mustafa;Nasir;Omar;Osman;Rahim;Rahman;Rais;Rashid;Ridzuan;Rosli;Saad;Sabri;Salleh;Samad;Shafie;Sharif;Sulaiman;Syed;Tahir;Umar;Usman;Wahab;Yahya;Yusof;Zahari;Zain;Zaki;Zamri;Zulkifli;Abdul;Adnan;Afiq;Aiman;Alif;Amir;Arif;Asyraf;Azman;Azmi;Badri;Faiz;Faris;Fikri;Fitri;Hafiz;Hakim;Halim;Hanif;Haris;Hasif;Haziq;Hazwan;Hisham;Iqbal;Irfan;Izzat;Jasni;Jefri;Kamarul;Khairi;Khairul;Luqman;Mazlan;Muhd;Naim;Najib;Nasrul;Nazmi;Rafiq;Rahmat;Ramli;Rasyid;Rizal;Roslan;Rusli;Saiful;Shahril;Syafiq;Syahril;Syukri;Wan;Zahid;Zairi;Zamani;Zul

Main.java

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.LinkedList;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner strInput = new Scanner(System.in);
        Scanner intInput = new Scanner(System.in);

        LinkedList<Friend> sList = new LinkedList<Friend>();

        System.out.println(
                "1. Insert records\n2. View record\n3. Remove the record\n4.
Update record\n5. Print all records\n6. Sort the records\n7. Generate 10
random data for linked list\n8. Exit\n");

        System.out.print("Enter your choice: ");
        int choice = intInput.nextInt();

        System.out.println();

        while (true) {
            switch (choice) {
                case 1:
                    System.out.print("Enter the number of friends: ");
                    int numFriends = intInput.nextInt();

                    if (numFriends < 1) {
                        System.out.println("Invalid number of friends.");
                        System.out.println();
                        break;
                    }

                    System.out.println();

                    for (int i = 0; i < numFriends; i++) {
                        System.out.print("Enter the ID number of friend " + (i
+ 1) + ": ");
                        int idno = intInput.nextInt();

                        System.out.print("Enter the name of friend " + (i + 1)
+ ": ");
                        String name = strInput.nextLine();
```

```
                        System.out.print("Enter the phone number of friend " +
(i + 1) + ": ");
                        String hpno = strInput.nextLine();

                        System.out.print("Enter the email of friend " + (i +
1) + ": ");
                        String email = strInput.nextLine();

                        // check if idno already exists without using contains
method
                        boolean idnoExists = false;
                        for (Friend friend : sList) {
                            if (friend.getIdno() == idno) {
                                idnoExists = true;
                                break;
                            }
                        }

                        if (idnoExists) {
                            System.out.println("The data already exists.
Please re-enter.");
                            System.out.println();
                            i--;
                            continue;
                        } else {
                            System.out.println();
                            System.out.println("The data has been inserted and
no duplicates exist.");
                            sList.add(i, new Friend(idno, name, hpno, email));
                        }

                        System.out.println();
                    }
                    break;
                case 2:
                    System.out.print("Enter the ID number of the friend to
view: ");
                    int idno = intInput.nextInt();

                    System.out.println();

                    boolean found = false;

                    // sort the list by idno using bubble sort
                    for (int i = 0; i < sList.size() - 1; i++) {
                        for (int j = 0; j < sList.size() - i - 1; j++) {
                            if (sList.get(j).getIdno() > sList.get(j +
1).getIdno()) {
```

```
                                Friend temp = sList.get(j);
                                sList.set(j, sList.get(j + 1));
                                sList.set(j + 1, temp);
                            }
                        }
                    }

                    // search for the friend with the given idno using binary
search
                    int low = 0;
                    int high = sList.size() - 1;
                    while (low <= high) {
                        int mid = (low + high) / 2;
                        if (sList.get(mid).getIdno() == idno) {
                            found = true;
                            System.out.println("ID number: " +
sList.get(mid).getIdno());
                            System.out.println("Name: " +
sList.get(mid).getName());
                            System.out.println("Phone number: " +
sList.get(mid).getHpno());
                            System.out.println("Email: " +
sList.get(mid).getEmail());
                            System.out.println();
                            break;
                        } else if (sList.get(mid).getIdno() < idno) {
                            low = mid + 1;
                        } else {
                            high = mid - 1;
                        }
                    }

                    if (!found) {
                        System.out.println("The data does not exist.");
                        System.out.println();
                    }
                    break;
                case 3:
                    // if the list is still empty, print out it is empty so
nothing can be removed
                    if (sList.size() == 0) {
                        System.out.println("The list is empty.");
                        System.out.println();
                        break;
                    }

                    System.out.print("Enter the index of the friend to remove
(0 to " + (sList.size() - 1) + "): ");
```

```
                    int index = intInput.nextInt();

                    System.out.println();

                    found = false;

                    for (int i = 0; i < sList.size(); i++) {
                        if (i == index) {
                            System.out.println("The data has been removed.");
                            sList.remove(i);
                            found = true;
                            break;
                        }
                    }

                    System.out.println();

                    if (!found) {
                        System.out.println("The data does not exist.");
                        System.out.println();
                    } else {

                        System.out.println("The list after removing the data:
");

                        // sort the list by idno using bubble sort
                        for (int i = 0; i < sList.size() - 1; i++) {
                            for (int j = 0; j < sList.size() - i - 1; j++) {
                                if (sList.get(j).getIdno() > sList.get(j +
1).getIdno()) {
                                    Friend temp = sList.get(j);
                                    sList.set(j, sList.get(j + 1));
                                    sList.set(j + 1, temp);
                                }
                            }
                        }

                        // print out the list after sorting
                        for (Friend friend : sList) {
                            System.out.println("ID number: " +
friend.getIdno());
                            System.out.println("Name: " + friend.getName());
                            System.out.println("Phone number: " +
friend.getHpno());
                            System.out.println("Email: " + friend.getEmail());
                            System.out.println();
                        }
                    }
```

```
                    break;
                case 4:
                    System.out.print("Enter the ID number of the friend to
update: ");
                    idno = intInput.nextInt();

                    System.out.println();

                    found = false;
                    for (Friend friend : sList) {
                        if (friend.getIdno() == idno) {

                            System.out.print("Enter the new phone number of
friend " + idno + ": ");
                            String hpno = strInput.nextLine();

                            System.out.print("Enter the new email of friend "
+ idno + ": ");
                            String email = strInput.nextLine();

                            sList.set(sList.indexOf(friend), new Friend(idno,
friend.getName(), hpno, email));

                            System.out.println("The data has been updated.");
                            System.out.println();
                            found = true;
                            break;
                        }
                    }

                    if (!found) {
                        System.out.println("The ID number does not exist.");
                        System.out.println();
                    }
                    break;
                case 5:
                    boolean isEmpty = true;
                    for (Friend friend : sList) {
                        System.out.println("ID number: " + friend.getIdno());
                        System.out.println("Name: " + friend.getName());
                        System.out.println("Phone number: " +
friend.getHpno());
                        System.out.println("Email: " + friend.getEmail());
                        System.out.println();
                        isEmpty = false;
                    }
```

```
                    if (isEmpty) {
                        System.out.println("The list is empty.");
                        System.out.println();
                    }
                    break;
                case 6:
                    // check if list is empty
                    if (sList.size() == 0) {
                        System.out.println("The list is empty.");
                        System.out.println();
                        break;
                    }

                    System.out.println("Sorting by ID number...");
                    for (int i = 0; i < sList.size(); i++) {
                        for (int j = 0; j < sList.size() - i - 1; j++) {
                            if (sList.get(j).getIdno() > sList.get(j +
1).getIdno()) {
                                Friend temp = sList.get(j);
                                sList.set(j, sList.get(j + 1));
                                sList.set(j + 1, temp);
                            }
                        }
                    }
                    System.out.println("The data has been sorted.");
                    System.out.println();
                    System.out.println("Updated list:");
                    for (Friend friend : sList) {
                        System.out.println("ID number: " + friend.getIdno());
                        System.out.println("Name: " + friend.getName());
                        System.out.println("Phone number: " +
friend.getHpno());
                        System.out.println("Email: " + friend.getEmail());
                        System.out.println();
                    }
                    break;
                case 7:
                    for (int i = 0; i < 10; i++) {

                        int id = (int) (Math.random() * 10000000) + 1;
                        String name = "";
                        // read from name.txt
                        BufferedReader br = null;
                        try {
                            br = new BufferedReader(new
FileReader("name.txt"));
                            String line = br.readLine();
                            int count = 0;
```

```
                            while (line != null) {
                                // split line by ;
                                String[] split = line.split(";");
                                // takes random name from name.txt
                                if (count == 0) {
                                    name += split[(int) (Math.random() *
split.length)] + " ";
                                    count++;
                                } else {
                                    name += split[(int) (Math.random() *
split.length)] + " ";
                                }

                                line = br.readLine();
                            }
                        } catch (IOException e) {
                            e.printStackTrace();
                        } finally {
                            try {
                                br.close();
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                        }

                        String hpno = "01" + (int) (Math.random() * 9) + "-" +
(int) (Math.random() * 1000000) + 1;
                        String email = name.replaceAll(" ", "").toLowerCase()
+ (int) (Math.random() * 100) + 1
                                + "@gmail.com";

                        // check if idno already exists
                        if (sList.contains(new Friend(id, name, hpno, email)))
{
                            i--;
                            continue;
                        } else {
                            sList.add(new Friend(id, name, hpno, email));
                        }
                    }
                    System.out.println("10 random data has been generated.");
                    System.out.println();
                    break;
                case 8:
                    System.out.println("Thank you for using this program.");
                    System.out.println();
                    System.exit(0);
                    break;
                default:
```

```
                    System.out.println("Invalid choice.");
                    break;

            }

            System.out.println("Press enter to continue...");
            strInput.nextLine();

            System.out.println(
                    "1. Insert records\n2. View record\n3. Remove the
record\n4. Update record\n5. Print all records\n6. Sort the records\n7.
Generate 10 random data for linked list\n8. Exit\n");

            System.out.print("Enter your choice: ");
            choice = intInput.nextInt();

            System.out.println();
        }
    }
}
```

Sample Input/Output

```
1. Insert records
2. View record
3. Remove the record
4. Update record
5. Print all records
6. Sort the records
7. Generate 10 random data for linked list
8. Exit

Enter your choice: 1

Enter the number of friends: 2

Enter the ID number of friend 1: 2022676488
Enter the name of friend 1: hazeeq
Enter the phone number of friend 1: 01111495803
Enter the email of friend 1: haikalroslan740@gmail.com

The data has been inserted and no duplicates exist.

Enter the ID number of friend 2: 2022676696
Enter the name of friend 2: khairul
Enter the phone number of friend 2: 0134265020
Enter the email of friend 2: chekhairul253@gmail.com

The data has been inserted and no duplicates exist.

Press enter to continue...
```

```
 1. Insert records
 2. View record
 3. Remove the record
 4. Update record
 5. Print all records
 6. Sort the records
 7. Generate 10 random data for linked list
 8. Exit

 Enter your choice: 2

 Enter the ID number of the friend to view: 2022676696

 ID number: 2022676696
 Name: khairul
 Phone number: 0134265020
 Email: chekhairul253@gmail.com
```

```
1. Insert records
2. View record
3. Remove the record
4. Update record
5. Print all records
6. Sort the records
7. Generate 10 random data for linked list
8. Exit

Enter your choice: 3

Enter the index of the friend to remove (0 to 1): 1

The data has been removed.

The list after removing the data:
ID number: 2022676488
Name: hazeeq
Phone number: 01111495803
Email: haikalroslan740@gmail.com

Press enter to continue...
```

```
1. Insert records
2. View record
3. Remove the record
4. Update record
5. Print all records
6. Sort the records
7. Generate 10 random data for linked list
8. Exit

Enter your choice: 4

Enter the ID number of the friend to update: 2022676488

Enter the new phone number of friend 2022676488: 0194269150
Enter the new email of friend 2022676488: hazeeqhaikal@gmail.com
The data has been updated.
```

```
1. Insert records
2. View record
3. Remove the record
4. Update record
5. Print all records
6. Sort the records
7. Generate 10 random data for linked list
8. Exit

Enter your choice: 5

ID number: 2022676488
Name: hazeeq
Phone number: 0194269150
Email: hazeeqhaikal@gmail.com
```

```
1. Insert records
2. View record
3. Remove the record
4. Update record
5. Print all records
6. Sort the records
7. Generate 10 random data for linked list
8. Exit

Enter your choice: 7

10 random data has been generated.
```

```
1. Insert records
2. View record
3. Remove the record
4. Update record
5. Print all records
6. Sort the records
7. Generate 10 random data for linked list
8. Exit

Enter your choice: 6

Sorting by ID number...
The data has been sorted.

Updated list:
ID number: 1334388
Name: Natasha Hazwan
Phone number: 010-8740781
Email: natashahazwan491@gmail.com

ID number: 3479105
Name: Julia Omar
Phone number: 010-6248461
Email: juliaomar831@gmail.com

ID number: 4206704
Name: Jamal Hisham
Phone number: 013-6444691
Email: jamalhisham131@gmail.com

ID number: 4797670
Name: Samad Jasni
Phone number: 010-9692611
Email: samadjasni141@gmail.com

ID number: 4974634
Name: Qistina Amir
Phone number: 017-6445321
Email: qistinaamir831@gmail.com
```

```
ID number: 5021132
Name: Balqis Faris
Phone number: 011-9780931
Email: balqisfaris821@gmail.com

ID number: 7504232
Name: Rosli Jalil
Phone number: 017-154261
Email: roslijalil761@gmail.com

ID number: 7527882
Name: Hamid Nasrul
Phone number: 018-7296671
Email: hamidnasrul441@gmail.com

ID number: 8171748
Name: Qistina Johan
Phone number: 013-8266881
Email: qistinajohan01@gmail.com

ID number: 8388225
Name: Rais Ridzuan
Phone number: 014-6733911
Email: raisridzuan81@gmail.com

ID number: 2022676488
Name: hazeeq
Phone number: 0194269150
Email: hazeeqhaikal@gmail.com
```

```
1. Insert records
2. View record
3. Remove the record
4. Update record
5. Print all records
6. Sort the records
7. Generate 10 random data for linked list
8. Exit

Enter your choice: 8

Thank you for using this program.
```