



## **CSC3202-1 : DATA STRUCTURE & ALGORITHM**

### **ASSIGNMENT 3**

#### **LECTURE GROUP : 1**

<b>NAME</b>	MUHAMMAD HAZUAN BIN SADIK
<b>MATRIC NO.</b>	223877
<b>PROGRAM</b>	BACHELOR OF COMPUTER SCIENCE WITH HONOURS
<b>EMAIL</b>	223877@student.upm.edu.my

## **Java Implementation:**

```
package NewAssignment3;

class Node {
    Profile profile;
    Node next;

    public Node(Profile profile) {
        this.profile = profile;
        this.next = null;
    }
}

class FriendNode {
    Profile friend;
    FriendNode next;

    public FriendNode(Profile friend) {
        this.friend = friend;
        this.next = null;
    }
}
```

```

package NewAssignment3;

class Profile {
    String name;
    String image;
    String status;
    FriendNode friendsHead;

    public Profile(String name, String image, String status) {
        this.name = name;
        this.image = image;
        this.status = status;
        this.friendsHead = null;
    }

    public void addFriend(Profile friend) {
        if (!isFriend(friend)) {
            FriendNode newFriend = new FriendNode(friend);
            newFriend.next = friendsHead;
            friendsHead = newFriend;
        }
    }

    public void removeFriend(Profile friend) {
        FriendNode current = friendsHead, prev = null;
        while (current != null && current.friend != friend) {
            prev = current;
            current = current.next;
        }
        if (current != null) {
            if (prev == null) {
                friendsHead = current.next;
            } else {
                prev.next = current.next;
            }
        }
    }

    public boolean isFriend(Profile friend) {
        FriendNode current = friendsHead;
        while (current != null) {
            if (current.friend == friend) {
                return true;
            }
            current = current.next;
        }
        return false;
    }

    @Override
    public String toString() {
        return "Name: " + name + "\nImage: " + (image.isEmpty() ? "No
image" : image) + "\nStatus: " + status;
    }
}

```

```
}  
  
public String listFriends() {  
    if (friendsHead == null) return "No friends yet.";  
    StringBuilder friendList = new StringBuilder("Friends:\n");  
    FriendNode current = friendsHead;  
    while (current != null) {  
        friendList.append(current.friend.name).append("\n");  
        current = current.next;  
    }  
    return friendList.toString();  
}  
}
```

```

package NewAssignment3;

class MukaBuku {
    Node head;

    public MukaBuku() {
        this.head = null;
    }

    public void addProfile(String name, String image, String status) {
        Profile newProfile = new Profile(name, image, status);
        Node newNode = new Node(newProfile);
        newNode.next = head;
        head = newNode;
    }

    public void removeProfile(String name) {
        Node current = head, prev = null;
        while (current != null && !current.profile.name.equalsIgnoreCase(
name)) {
            prev = current;
            current = current.next;
        }
        if (current != null) {
            if (prev == null) {
                head = current.next;
            } else {
                prev.next = current.next;
            }
            Node temp = head;
            while (temp != null) {
                temp.profile.removeFriend(current.profile);
                temp = temp.next;
            }
        }
    }

    public Profile findProfile(String name) {
        Node current = head;
        while (current != null) {
            if (current.profile.name.equalsIgnoreCase(name)) {
                return current.profile;
            }
            current = current.next;
        }
        return null;
    }

    public void displayProfiles() {
        if (head == null) {
            System.out.println("No profiles in the network.");
            return;
        }
    }
}

```

```
Node current = head;
while (current != null) {
    System.out.println(current.profile);
    System.out.println(current.profile.listFriends());
    System.out.println("-----");
    current = current.next;
}
}
```

```

package NewAssignment3;

import java.util.Scanner;

public class MukaBukuApp {
    public static void main(String[] args) {
        MukaBuku network = new MukaBuku();
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n===== LAMAN MUKA BUKU =====");
            System.out.println("===== by HAZUAN =====");
            System.out.println("\n" +
                "+-----+-----+\n" +
                "| Option | Function | \n" +
                "+-----+-----+\n" +
                "| 1 | Create Profile | \n" +
                "| 2 | Delete Profile | \n" +
                "| 3 | Search Profile | \n" +
                "| 4 | Modify Profile | \n" +
                "| 5 | Add Friend | \n" +
                "| 6 | Display Profiles | \n" +
                "| 7 | Exit | \n" +
                "+-----+-----+\n");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter name: ");
                    String name = scanner.nextLine();
                    System.out.print("Enter image path: ");
                    String image = scanner.nextLine();
                    System.out.print("Enter status: ");
                    String status = scanner.nextLine();
                    network.addProfile(name, image, status);
                    break;

                case 2:
                    System.out.print("Enter name of profile to delete: ")

                    name = scanner.nextLine();
                    network.removeProfile(name);
                    break;

                case 3:
                    System.out.print("Enter name to search: ");
                    name = scanner.nextLine();
                    Profile profile = network.findProfile(name);
                    if (profile != null) {
                        System.out.println(profile);
                    }
                }
            }
        } while (choice != 7);
    }
}

```



```

        System.out.println(profile.ListFriends());
    } else {
        System.out.println("Profile not found.");
    }
    break;

case 4:
    System.out.print("Enter name of profile to modify: ");

    name = scanner.nextLine();
    profile = network.findProfile(name);
    if (profile != null) {
        System.out.print("Enter new image path (leave
blank to keep current): ");
        image = scanner.nextLine();
        if (!image.isEmpty()) profile.image = image;

        System.out.print("Enter new status (leave blank
to keep current): ");
        status = scanner.nextLine();
        if (!status.isEmpty()) profile.status = status;
    } else {
        System.out.println("Profile not found.");
    }
    break;

case 5:
    System.out.print("Enter your name: ");
    String yourName = scanner.nextLine();
    Profile yourProfile = network.findProfile(yourName);
    if (yourProfile != null) {
        System.out.print("Enter friend's name: ");
        String friendName = scanner.nextLine();
        Profile friendProfile = network.findProfile(
friendName);

        if (friendProfile != null) {
            yourProfile.addFriend(friendProfile);
        } else {
            System.out.println("Friend's profile not
found.");
        }
    } else {
        System.out.println("Your profile not found.");
    }
    break;

case 6:
    network.displayProfiles();
    break;

case 7:
    System.out.println("Exiting application.");
    break;

```

```
        default:
            System.out.println("Invalid choice. Try again.");
        }
    } while (choice != 7);

    scanner.close();
}
}
```