



Student ID: 0789055

Student Name: Dikshith Reddy Macherla

Program Testing: Screenshot based functionality testing with a short description of each test. You can use the same format for testing that you used in COIS 1020H.

```
PS C:\Users\diksh\Desktop\COIS 2240H\Lab_2\COIS2240Lab2> & 'C:\Program Files\Amazon Corretto\jdk21.0.2_13\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,address=localhost:54819' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\diksh\AppData\Roaming\Code\User\workspaceStorage\8a58a9af32d6dcae2cbc85a\redhat.java\jdt_ws\COIS2240Lab2_ae61ba5a\bin' 'Program'
Listing all contacts:
Contact{name='Aladdin', email='aladdin@example.com', phoneNumber='123-456-7890'}
Contact{name='Bob', email='bob@example.com', phoneNumber='098-765-4321'}

Removing Aladdin and attempting to remove non-existing contact:
Contact not found!

Listing all contacts after removal:
Contact{name='Bob', email='bob@example.com', phoneNumber='098-765-4321'}

Listing all contacts after adding Charlie:
Contact{name='Bob', email='bob@example.com', phoneNumber='098-765-4321'}
Contact{name='Charlie', email='charlie@example.com', phoneNumber='321-654-0987'}
PS C:\Users\diksh\Desktop\COIS 2240H\Lab_2\COIS2240Lab2>
```

Copy and paste your code below:

Program.java

```
public class Program {
    public static void main(String[] args) {
        ContactManager manager = new ContactManager();

        // Add contacts
        manager.addContact(new Contact("Aladdin", "aladdin@example.com", "123-456-7890"));
        manager.addContact(new Contact("Bob", "bob@example.com", "098-765-4321"));

        // List contacts
        System.out.println("Listing all contacts:");
        manager.listContacts();

        // Remove a contact and show attempt to remove non-existing contact
    }
}
```

```

        System.out.println("\nRemoving Aladdin and attempting to remove non-
existing contact:");
        manager.removeContact("Aladdin");
        manager.removeContact("Charlie"); // This contact does not exist

        // List contacts again to see the change
        System.out.println("\nListing all contacts after removal:");
        manager.listContacts();

        // Add another contact and list again
        manager.addContact(new Contact("Charlie", "charlie@example.com", "321-
654-0987"));
        System.out.println("\nListing all contacts after adding Charlie:");
        manager.listContacts();
    }
}

```

ContactManager.java

```

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

// Define the ContactManager class
class ContactManager {
    // List to store Contact objects
    private List<Contact> contacts;

    // Constructor to initialize the ContactManager
    public ContactManager() {
        contacts = new ArrayList<>();
    }

    // Method to add a new Contact
    public void addContact(Contact contact) {
        contacts.add(contact);
    }

    // Method to remove a Contact by name
    public void removeContact(String name) {
        Contact contactToRemove = null;
        for (Contact contact : contacts) {

```

```

        if (contact.getName().equals(name)) {
            contactToRemove = contact;
            break;
        }
    }
    if (contactToRemove != null) {
        contacts.remove(contactToRemove);
    } else {
        System.out.println("Contact not found!");
    }
}

// Method to list all Contacts
public void listContacts() {
    for (Contact contact : contacts) {
        System.out.println(contact);
    }
}
}

```

Contact.java

```

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

// Define the Contact class
class Contact {
    // Properties of the Contact class
    private String name;
    private String email;
    private String phoneNumber;

    // Constructor to initialize a new Contact object
    public Contact(String name, String email, String phoneNumber) {
        this.name = name;
        this.email = email;
        this.phoneNumber = phoneNumber;
    }

    // Implement the toString method for Contact
    @Override
    public String toString() {

```

```

        return "Contact{" +
            "name='" + name + '\'' +
            ", email='" + email + '\'' +
            ", phoneNumber='" + phoneNumber + '\'' +
            '}'
    }

    public String getName() {
        return name;
    }

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

```

Prove using a screenshot that you cloned the github repository using the tool with VS Code (or the IDE that you picked instead of VS Code):

The screenshot shows the VS Code editor with a Java file named `Program.java` open. The code defines a `Contact` class and a `ContactManager` class. The `main` method in `Program` demonstrates the functionality of the `ContactManager` by adding, listing, and removing contacts. The output window at the bottom shows the execution results, including the list of contacts before and after removal and addition.

```

1 public class Program {
2     public static void main(String[] args) {
3         ContactManager manager = new ContactManager();
4
5         // Add contacts
6         manager.addContact(new Contact(name="Aladdin", email="aladdin@example.com", phoneNumber="123-456-7890"));
7         manager.addContact(new Contact(name="Bob", email="bob@example.com", phoneNumber="098-765-4321"));
8
9         // List contacts
10        System.out.println("Listing all contacts:");
11        manager.listContacts();
12
13        // Remove a contact and show attempt to remove non-existing contact
14        System.out.println("Removing Aladdin and attempting to remove non-existing contact:");
15        manager.removeContact(name="Aladdin");
16        manager.removeContact(name="Charlie"); // This contact does not exist
17
18        // List contacts again to see the change
19        System.out.println("Listing all contacts after removal:");
20        manager.listContacts();
21
22        // Add another contact and list again
23        manager.addContact(new Contact(name="Charlie", email="charlie@example.com", phoneNumber="321-654-0987"));
24        System.out.println("Listing all contacts after adding Charlie:");
25        manager.listContacts();
26    }
27 }
28

```

Output:

```

Removing Aladdin and attempting to remove non-existing contact:
Contact not found

Listing all contacts after removal:
Contact(name='Bob', email='bob@example.com', phoneNumber='098-765-4321')

Listing all contacts after adding Charlie:
Contact(name='Bob', email='bob@example.com', phoneNumber='098-765-4321')
Contact(name='Charlie', email='charlie@example.com', phoneNumber='321-654-0987')
PS C:\Users\diksh\Desktop\COIS 2240H(Lab_2)\COIS2240Lab2>

```

Prove using a screenshot that you successfully created a pull request:

