

United International University

Department of Computer Science & Engineering



Report On

“Object Oriented Programming Laboratory”

Course Code: CSE 1116 | Sec: B

Submitted to:

FmSh Fahim Shahriar

shahriar@cse.uiu.ac.bd

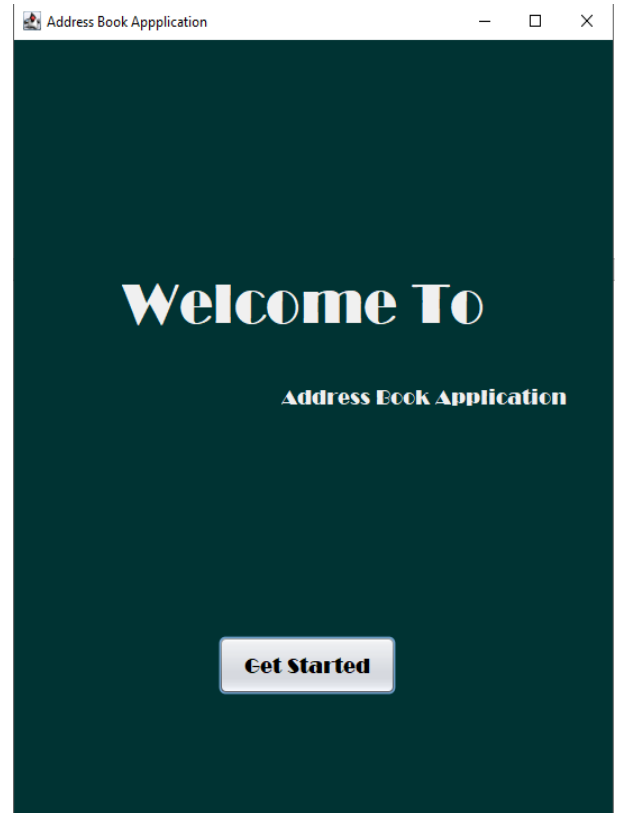
Submitted by:

Name: MD MEHEDI HASAN

ID: 011 201 292

Welcome Page

```
private void getStartedActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new OptionPage().setVisible(true);  
}  
  
/**  
 * @param args the command line arguments  
 */  
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    Look and feel setting code (optional)  
  
    /* Create and display the form */  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() { new WelcomePage().setVisible(true); }  
    });  
}
```



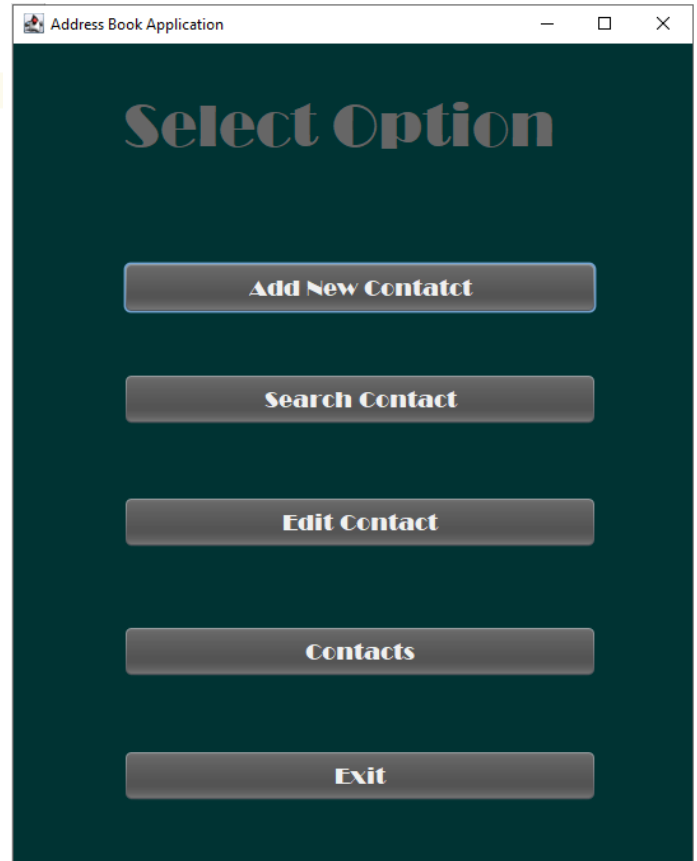
Working Detail

This GUI screenshot is the welcome page of Address Book Application. There is a button named **Get Started**.

- **Get Started:** By clicking this button user can go to the next page of this Application.

Select Option

```
private void addPeopleActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new AddNewContact().setVisible(true);  
}  
  
private void searchPeople2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new SearchContact().setVisible(true);  
}  
  
private void editPeople1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new EditContact().setVisible(true);  
}  
  
private void exitActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new Contacts().setVisible(true);  
}  
  
private void allPeople1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new Exit().setVisible(true);  
}
```



Working Detail

This GUI screenshot is the Select Option page of the Address Book Application. There are five buttons-

- **Add New Contact** : This Button is for adding new contact.
- **Search Contact** : User can search contact via clicking this button.
- **Edit Contact** : This button is for editing contact.
- **Contacts** : User can see all the contacts from here.
- **Exit** : User can exit from here clicking this button.

1. ADD NEW CONTACT

```
private void newAddressPreviousActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new OptionPage().setVisible(true); }  
  
private void newAddressPrevious1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if( ( nametf1.getText().equals("") ) ||  
        (phonetf1.getText().equals("") )  
        || (emailtf.getText().equals("") ) ||  
        (addresstf.getText().equals("") ) ) {}  
    else{  
        try {  
            FileWriter file;  
            file = new FileWriter( fileName: "File/A.txt", append: true);  
            PrintWriter write = new PrintWriter(file);  
            write.println(nametf1.getText()+"-"+phonetf1.getText()+"-"+  
                +emailtf.getText()+"-"+addresstf.getText());  
            write.close();  
            this.dispose();  
            new OptionPage().setVisible(true);  
        }  
        catch (IOException s ){  
            s.printStackTrace();  
        }  
    }  
}
```

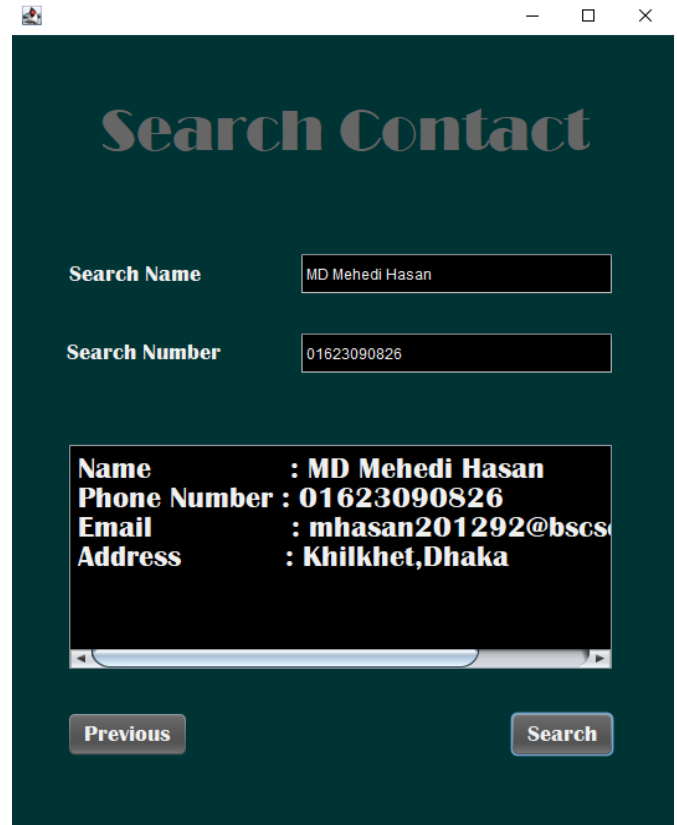
The screenshot shows a window titled "Address Book Application" with a sub-header "Add New Contact". The window has a dark teal background. It contains four text input fields with labels to their left: "Name" (containing "MD Mehedi Hasan"), "Phone No" (containing "01623090826"), "Email" (containing "mhasan201292@bscse.uju.ac.bd"), and "Street Address" (containing "Khilkhet,Dhaka"). At the bottom of the window, there are two buttons: "Previous" on the left and "Save" on the right.

Working Detail

From here, user can put input in the TextField according to the label. Here we can see two button. One is **Previous** and another is **Save**.

- **Save**: After filling up all the TextField when user click the Save button all the data/input from the TextField save into a ".txt" file and the application page return back to Select Option page. But if user won't fill all the TextField then this button will not work.
- **Previous**: By pressing this button user can return back to the previous page(Select Option Page).

2. Search Contact



A screenshot of a web application window titled "Search Contact". The window has a dark teal background. At the top, the title "Search Contact" is displayed in a large, bold, white serif font. Below the title, there are two input fields: "Search Name" with the value "MD Mehedi Hasan" and "Search Number" with the value "01623090826". Below these fields, a white box contains the search results in a bold, black serif font: "Name : MD Mehedi Hasan", "Phone Number : 01623090826", "Email : mhasan201292@bscsc", and "Address : Khilkhet,Dhaka". At the bottom of the window, there are two buttons: "Previous" and "Search", both in a light gray color with a dark gray border.

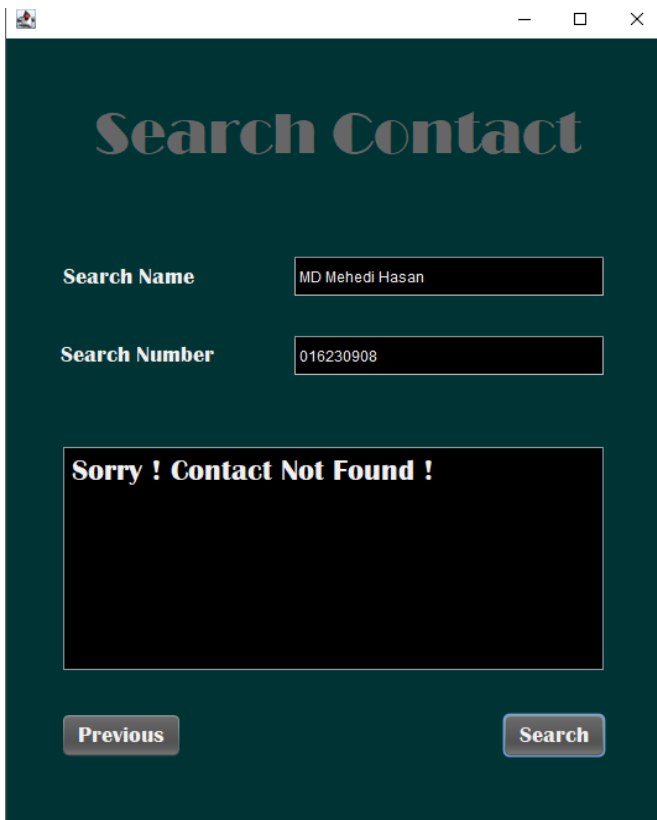
Search Contact

Search Name MD Mehedi Hasan

Search Number 01623090826

Name : MD Mehedi Hasan
Phone Number : 01623090826
Email : mhasan201292@bscsc
Address : Khilkhet,Dhaka

Previous **Search**



A screenshot of the same "Search Contact" web application window. The search inputs are the same: "Search Name" is "MD Mehedi Hasan" and "Search Number" is "016230908". However, the search results box now displays the message "Sorry ! Contact Not Found !" in a bold, black serif font. The "Previous" and "Search" buttons remain at the bottom.

Search Contact

Search Name MD Mehedi Hasan

Search Number 016230908

Sorry ! Contact Not Found !

Previous **Search**

```

private void searchContactPreviousActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    new OptionPage().setVisible(true);
}

private void searchContactPreviousActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String name = nameTF.getText();
    String number = numberTF.getText();
    int i, j = 0;
    try{
        ArrayList<People> array = new ArrayList<>();
        FileReader file = new FileReader("File/A.txt");
        BufferedReader r = new BufferedReader(file);
        String line;
        searchContactArea.setText("");
        while ( (line = r.readLine()) != null ){
            String[] part = line.split("-");
            People people = new People(part[0],part[1],part[2],part[3]);
            array.add(people);
        }
        for ( i = 0 ; i < array.size() ; i++ ){
            if( name.equals(array.get(i).name) && number.equals(array.get(i).phone_Number) ){
                searchContactArea.append("Name          : " +array.get(i).name + "\n"
                    + "Phone Number : " +array.get(i).phone_Number + "\n"
                    + "Email          : " + array.get(i).email + "\n"
                    + "Address         : " + array.get(i).address + "\n\n");
            }
            else{j++; }
        }

        if ( j == array.size() ){
            searchContactArea.setText("Sorry ! Contact Not Found !");
        }
        r.close();
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Contacts.class.getName()).log(Level.SEVERE, null, ex);
    } catch (java.io.IOException ex) {
        Logger.getLogger(Contacts.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Working Detail

Searching by peoples name and phone number user can see their Information details such as- name, phone number, email & address. There are also two buttons here such as-

- **Search** : By giving name and phone number in the search bar TextField and pressing the search button user can see peoples information detail in the TextArea. But if the given information is not matched with the save data, then In the TextArea there will show a message and that is “Sorry ! Contact Not Found !”
- **Previous** : This button will take the user in the previous page (Select Option Page).

3. Edit Contact

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    new OptionPage().setVisible(true);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String name = jTextField1.getText();
    String number = jTextField2.getText();
    int i, j = 0 ;
    try{
        ArrayList<People> array = new ArrayList<>();
        FileReader file = new FileReader( fileName: "File/A.txt");
        BufferedReader r = new BufferedReader(file);
        String line ;
        while ( (line = r.readLine() ) != null ){
            String[] part = line.split( regex: "-");
            People people = new People(part[0],part[1],part[2],part[3]) ;
            array.add(people);}
        for ( i = 0 ; i < array.size() ; i++ ){
            if( ( jTextField4.getText().equals("") ) || (jTextField5.getText().equals("") ) ||
                (jTextField6.getText().equals("") ) || (jTextField3.getText().equals("") ) ) {}
            else{
                if( name.equals(array.get(i).name) && number.equals(array.get(i).phone_Number) ){
                    array.get(i).name = jTextField4.getText() ;
                    array.get(i).phone_Number = jTextField5.getText() ;
                    array.get(i).email = jTextField6.getText() ;
                    array.get(i).address = jTextField3.getText() ;
                    break ;
                }
            }
        }
        FileWriter f;
        f = new FileWriter( fileName: "File/A.txt");
        PrintWriter w = new PrintWriter(f);
        for ( i = 0 ; i < array.size() ; i++ ){
            w.println(array.get(i).name+"-"+array.get(i).phone_Number+"-"+array.get(i).email+"-"+array.get(i).address );
        }
        this.dispose();
        new OptionPage().setVisible(true);
        r.close();
        w.close();
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Contacts.class.getName()).log(Level.SEVERE, msg: null, ex);
    } catch (java.io.IOException ex) {
        Logger.getLogger(Contacts.class.getName()).log(Level.SEVERE, msg: null, ex);
    }
}
```

Address Book Application

Edit Contact

Search Name MD Mehedi Hasan Search Number 01623090826

Name MD Mehedi

Phone No 01714336568

Email mhasan@gmail.com

Street Address Dhaka

Previous Save

Working Detail

User can edit contact from here by searching their name and phone number. If searching name and number matched with the save data then by filling up the TextField according to label user can edit contact.

- **Save** : This button saved the edited information in the “.txt” file.
- **Previous** : This button will take the user in the previous page (Select Option Page).

4. Contacts

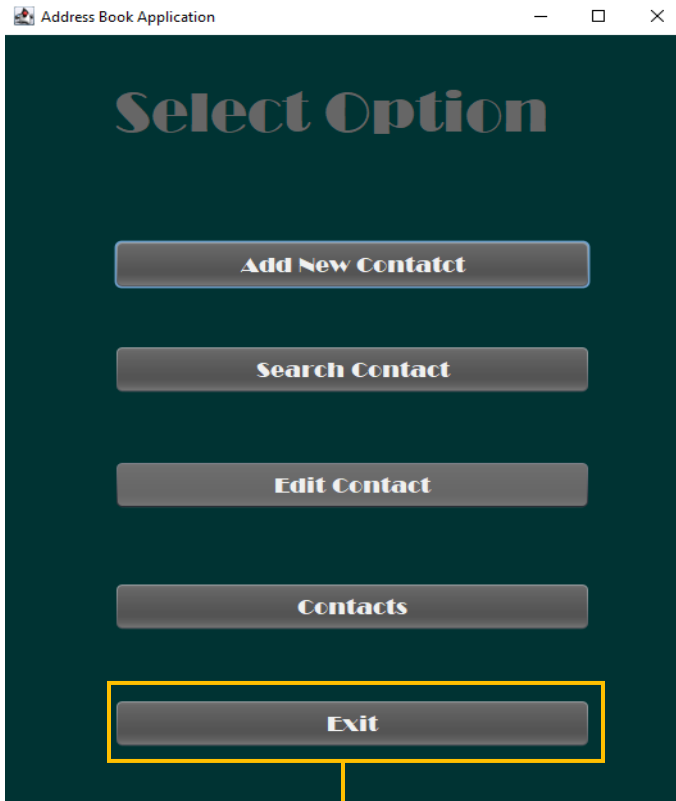
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    this.dispose();  
    new OptionPage().setVisible(true);  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{  
        ArrayList<People> array = new ArrayList<>();  
        FileReader file = new FileReader( fileName: "File/A.txt");  
        BufferedReader r = new BufferedReader(file);  
        String line ;  
        JTextArea1.setText("");  
        while ( (line = r.readLine() ) != null ){  
            String[] part = line.split( regex: "-");  
            People people = new People(part[0],part[1],part[2],part[3]) ;  
            array.add(people);  
        }  
        array.sort(new Comparator<People>() {  
            @Override public int compare(People o1, People o2) {  
                return o1.name.compareTo(o2.name);  
            }  
        });  
        for ( int i = 0 ; i < array.size() ; i++){  
            JTextArea1.append("Name           : " +array.get(i).name + "\n"  
                +"Phone Number : " +array.get(i).phone_Number + "\n"  
                +"Email           : " + array.get(i).email + "\n"  
                +"Address          : " + array.get(i).address + "\n\n");  
        }  
        r.close();  
    } catch (FileNotFoundException ex) {  
        Logger.getLogger(Contacts.class.getName()).log(Level.SEVERE, msg: null, ex);  
    } catch (IOException ex) {  
        Logger.getLogger(Contacts.class.getName()).log(Level.SEVERE, msg: null, ex);  
    }  
}
```



Working Detail

User Can see from here all the contact which they have saved already in ".txt" file and also the edited details which have been edited in Edit Contact Page by pressing the **Show Data** button. All the contact appear in the TextArea in ascending sorted order according to their name.

5.Exit



```
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {...} // </editor-fold>

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() { new Exit().setVisible(true); }
    });
}
```

