Assignment No. 01

Que 01 :- Installation of JDK Environment and following utilities. What is javac, javap and Javadoc.

Que 02 :- Design the application by using array.

Que 03 :- Implementation of Package, Interface and abstract class.

Que 04 :- Design application using string, string builder and string tokenizer.

Que 05 :- Test any five of standard exception and user defined custom exceptions in java.

Que 06 :- Threads creation and design application by using extending the thread class/Implementing the Runnable Interface. Application of multithreading in java.

Que 07 :- Design java application using collection in java such as array list , Link list.

Que 08 :- Design GUI based java application using AWT, Swing with Event Handling.

Que 09 :- Design and Implement JDBC Applications.

Que 10 :- Design and Implement servlet Application.

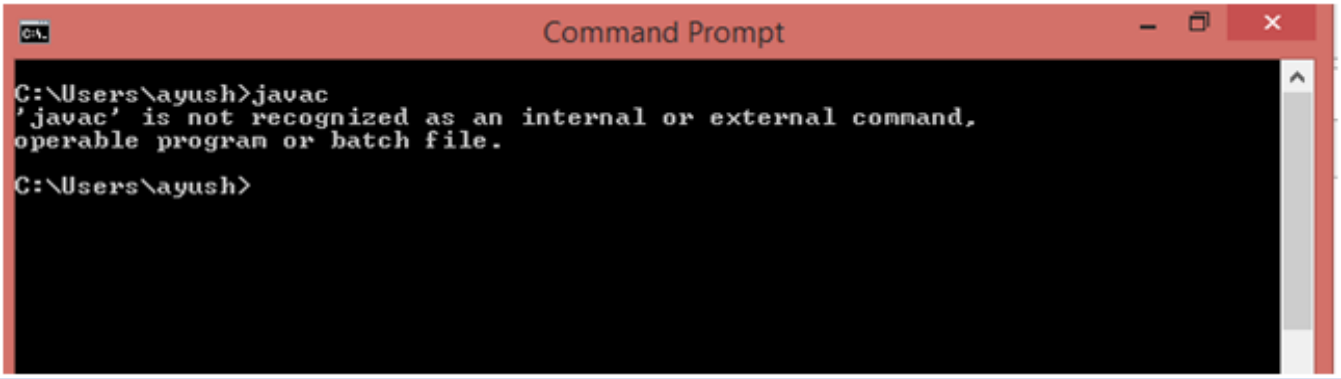
Que 11 :- Design and Implement JSP Application

Que 01 :- Installation of JDK Environment and following utilities. What is javac, javap and Javadoc.

Installation Process of JDK Environment :-

- Verify that it is already installed or not :-

  Check whether Java is already installed on the system or not. In my case, it is not installed therefore I need to install JDK 16 on my computer.



- Download JDK Software :-

  Click the below link to download jdk 16 for you windows 64 bit system.

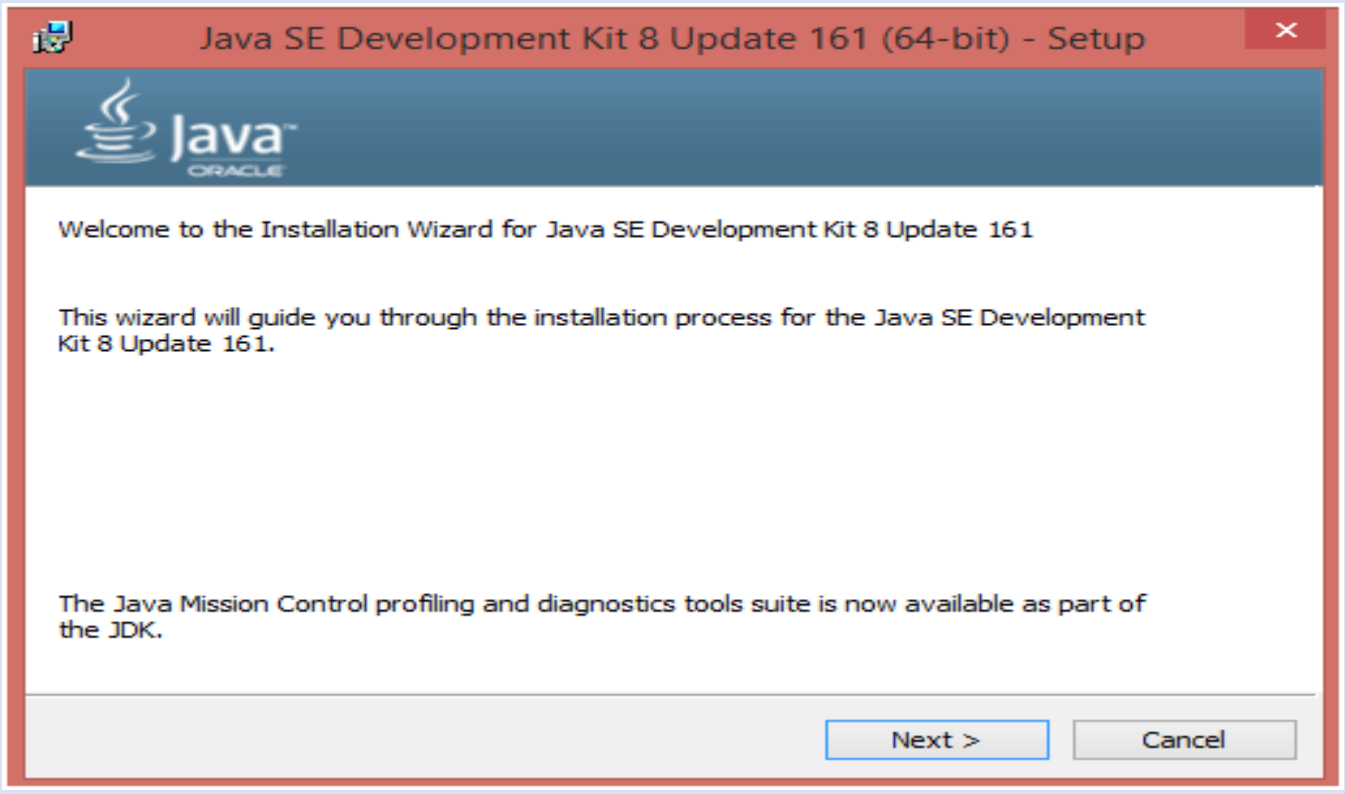
  Download JDK For Windows

  There are available releases for Linux and mac operating systems. You can visit the official link for JDK distributions i.e.

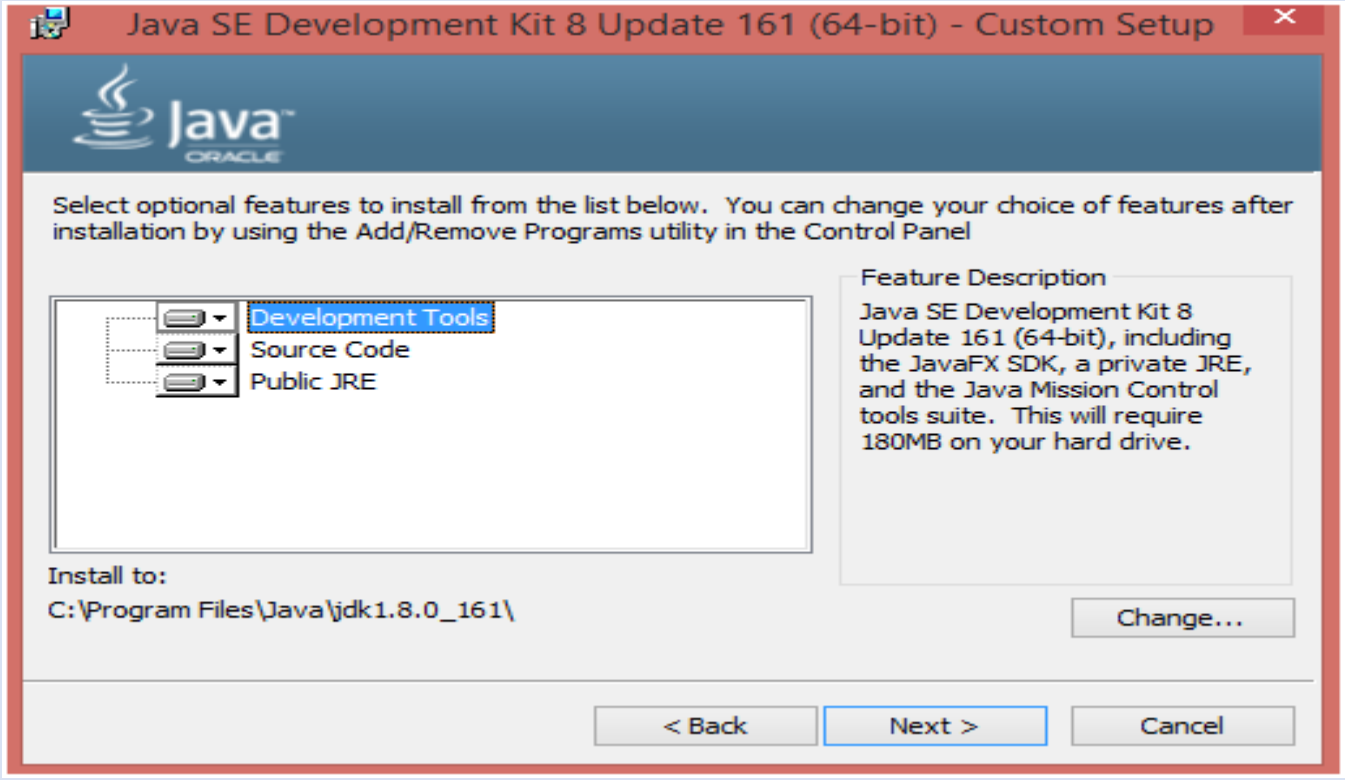
  JDK Downloads
  https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html

- Install JDK :-

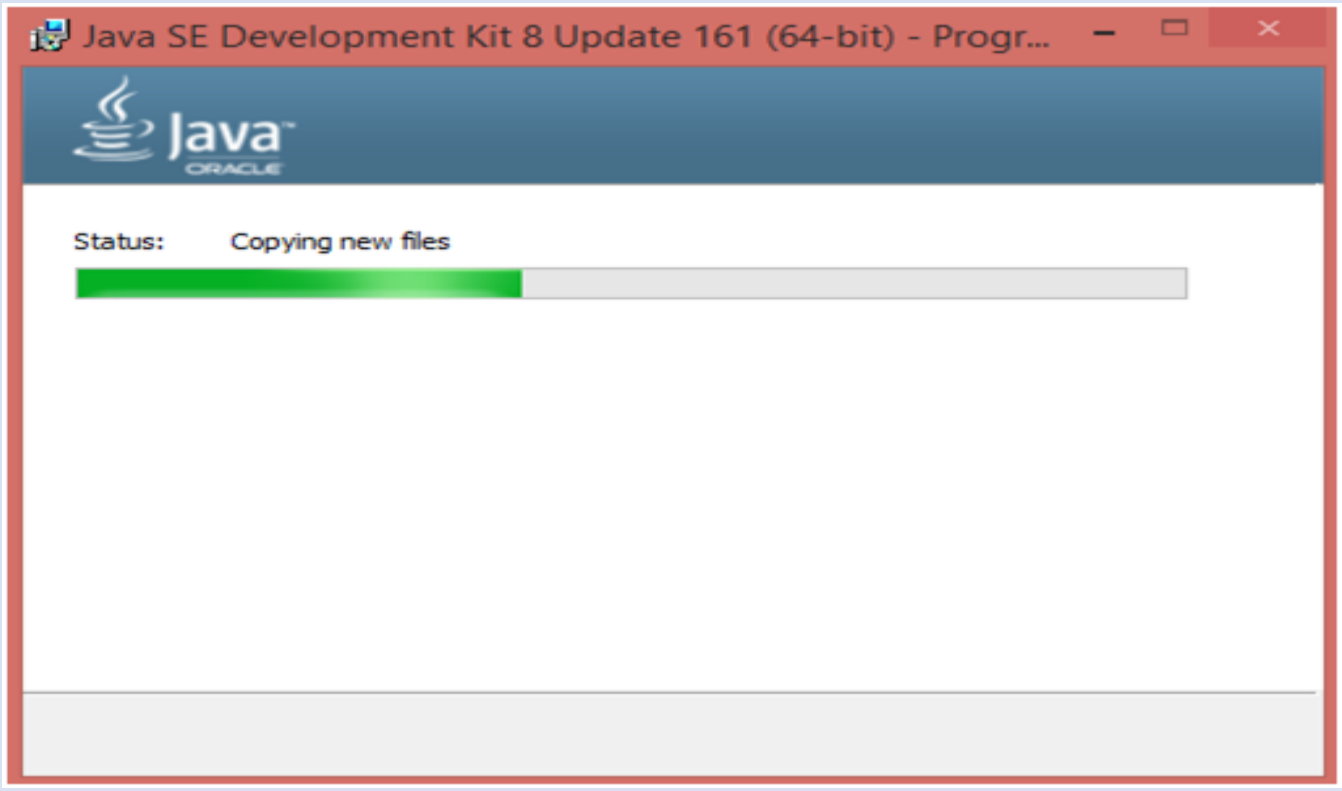
  Open the executable file which you have just downloaded and follow the steps.



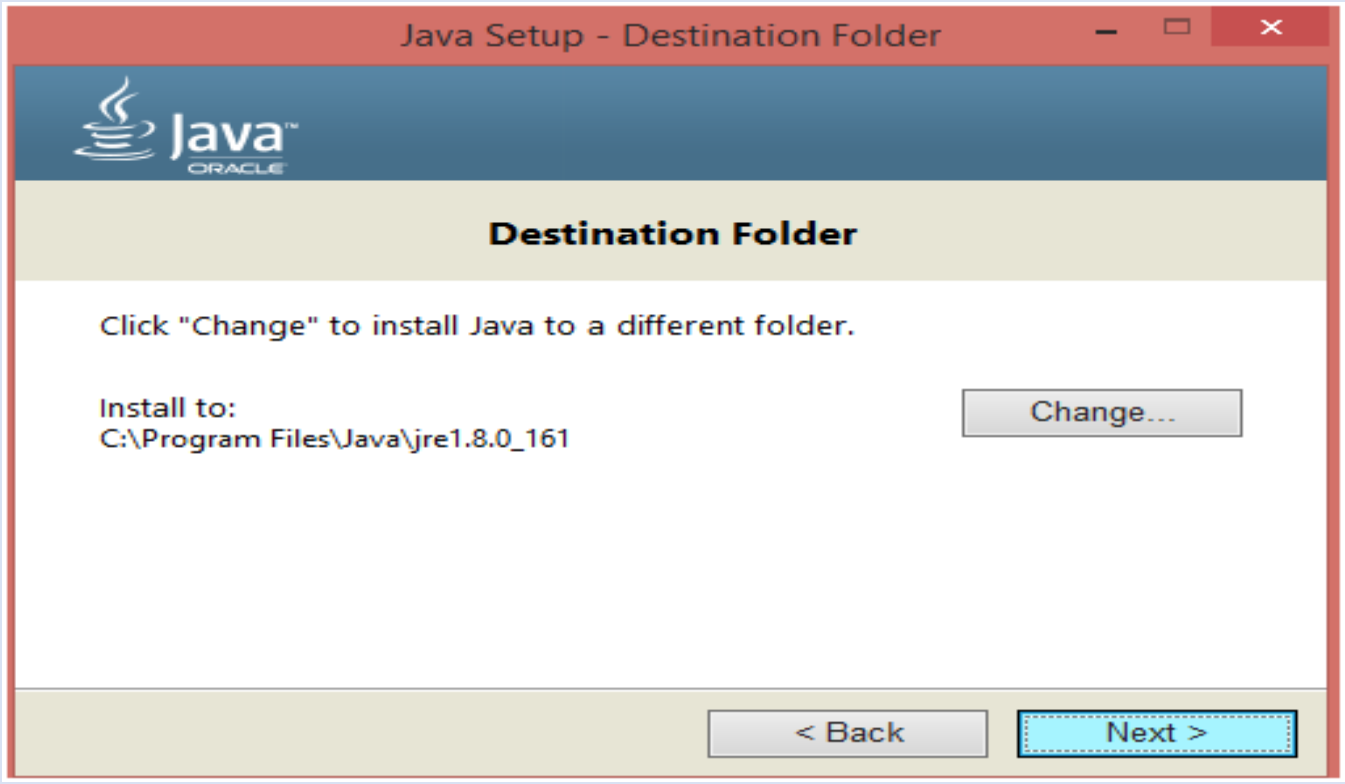
Click Next to continue



Just Choose Development Tools and click Next.

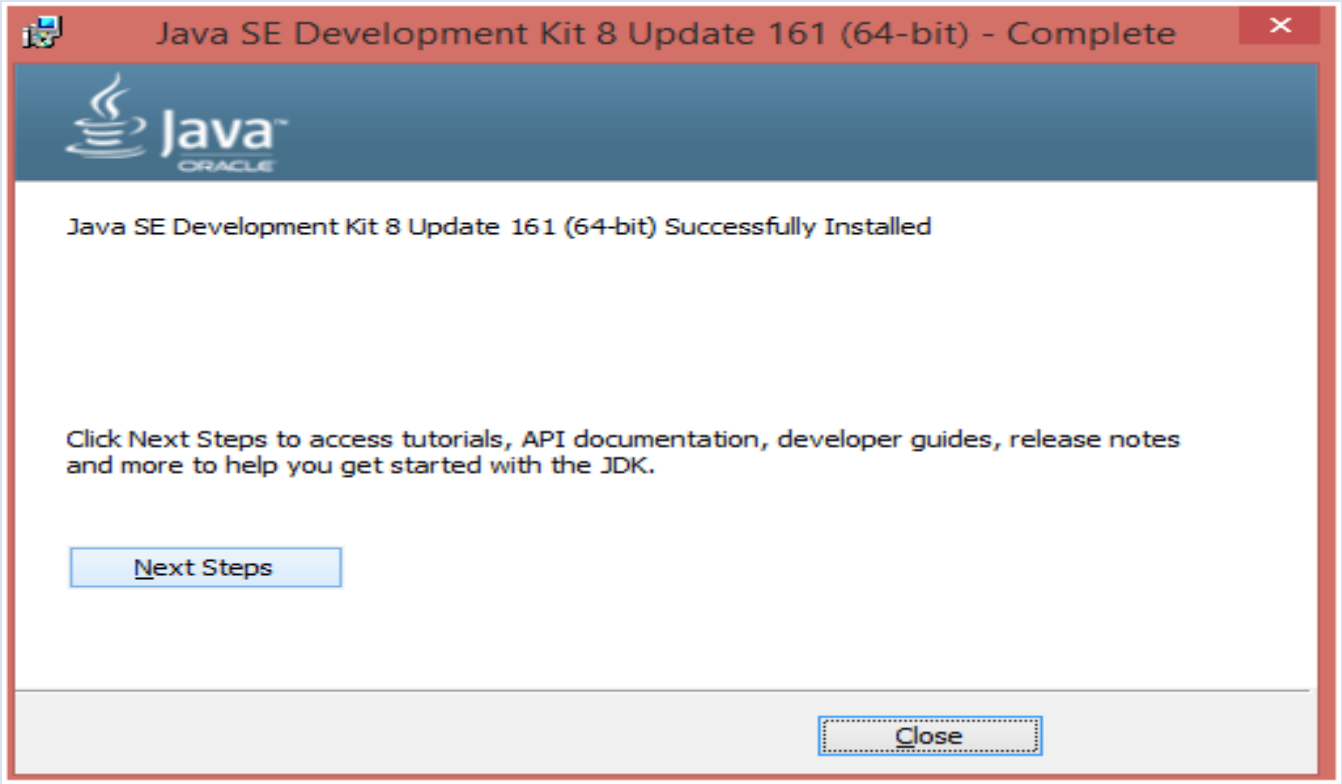


Set up is being ready.



Choose the Destination folder in which you want to install JDK. Click Next to continue with the installation.

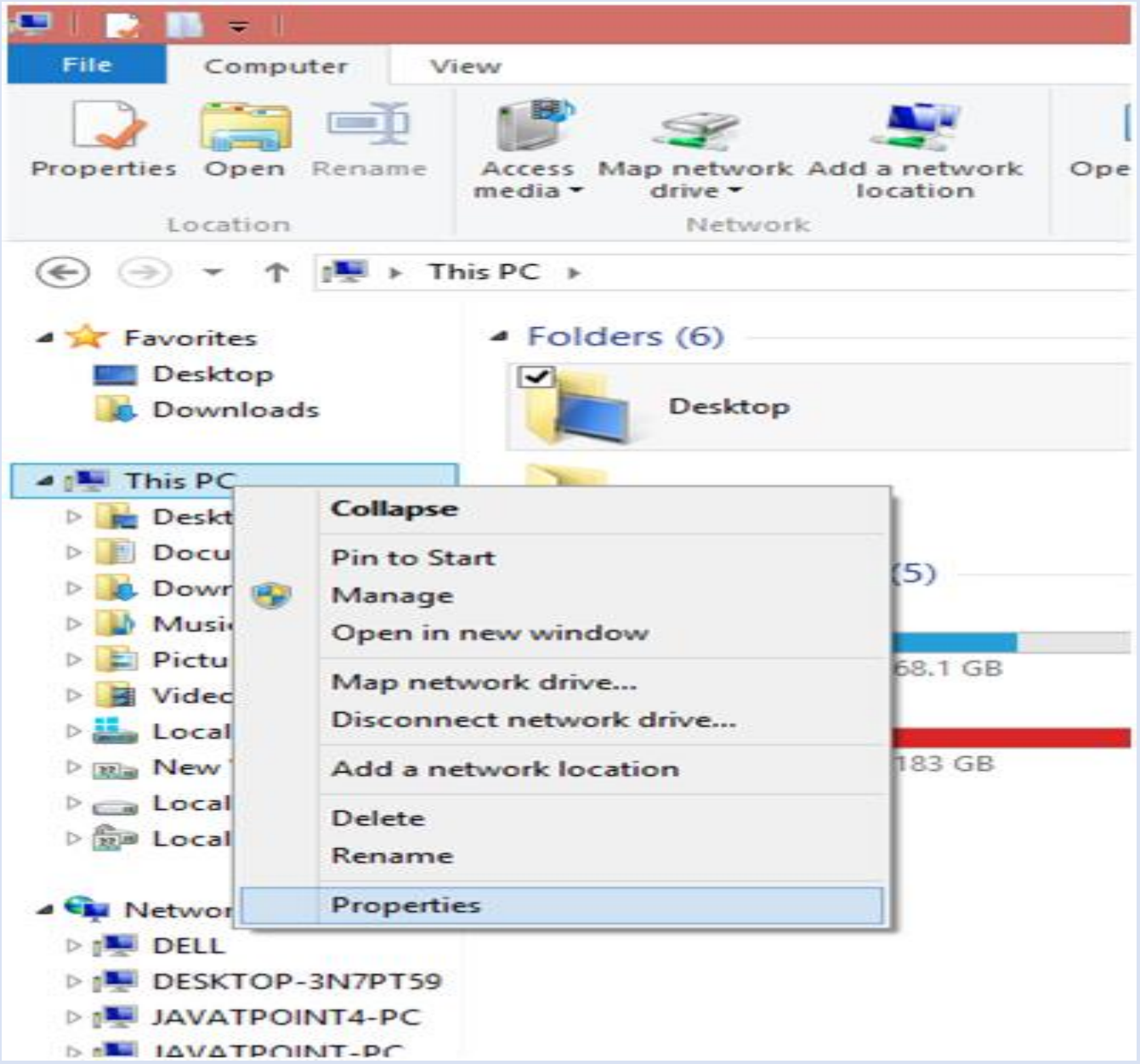
Set up is installing Java to the computer.



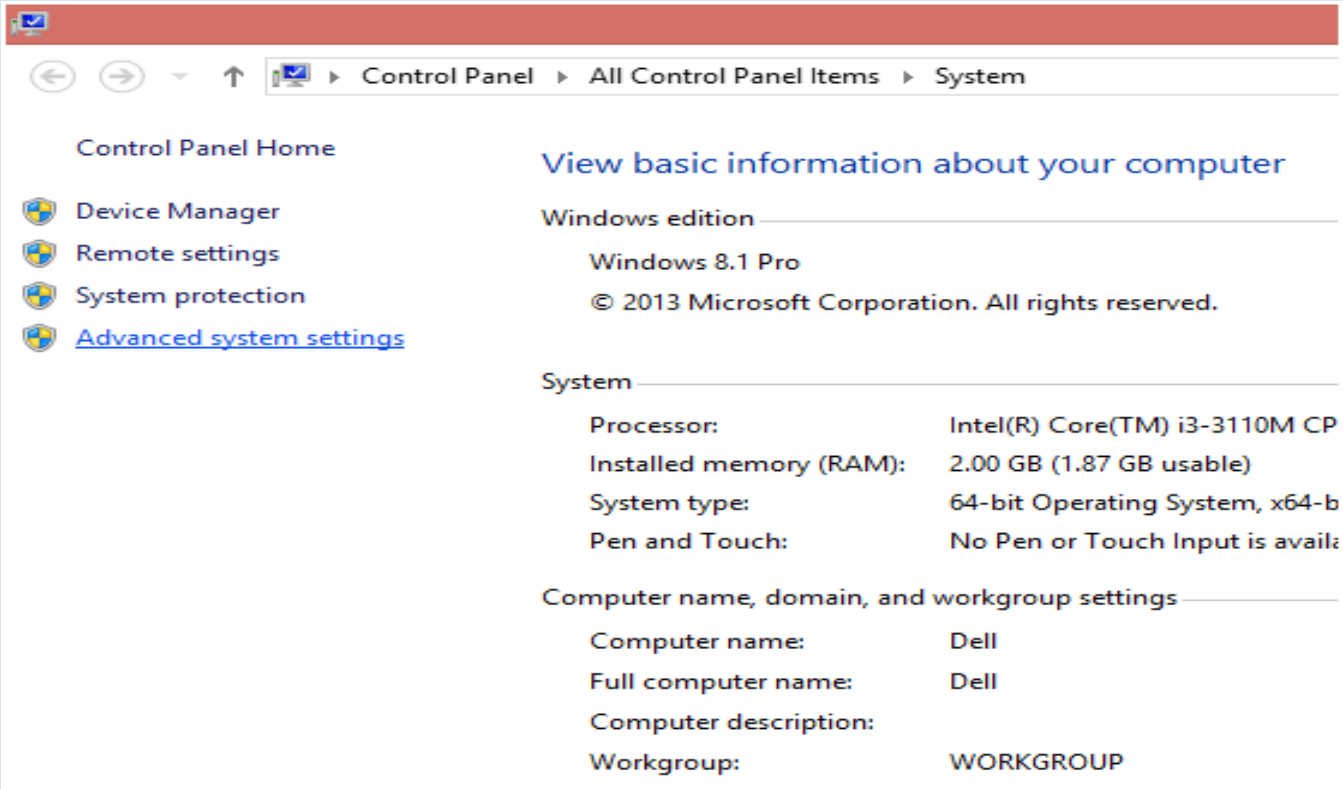
We have successfully installed Java SE development kit 8. Close the installation set up.

Set the Permanent Path :-

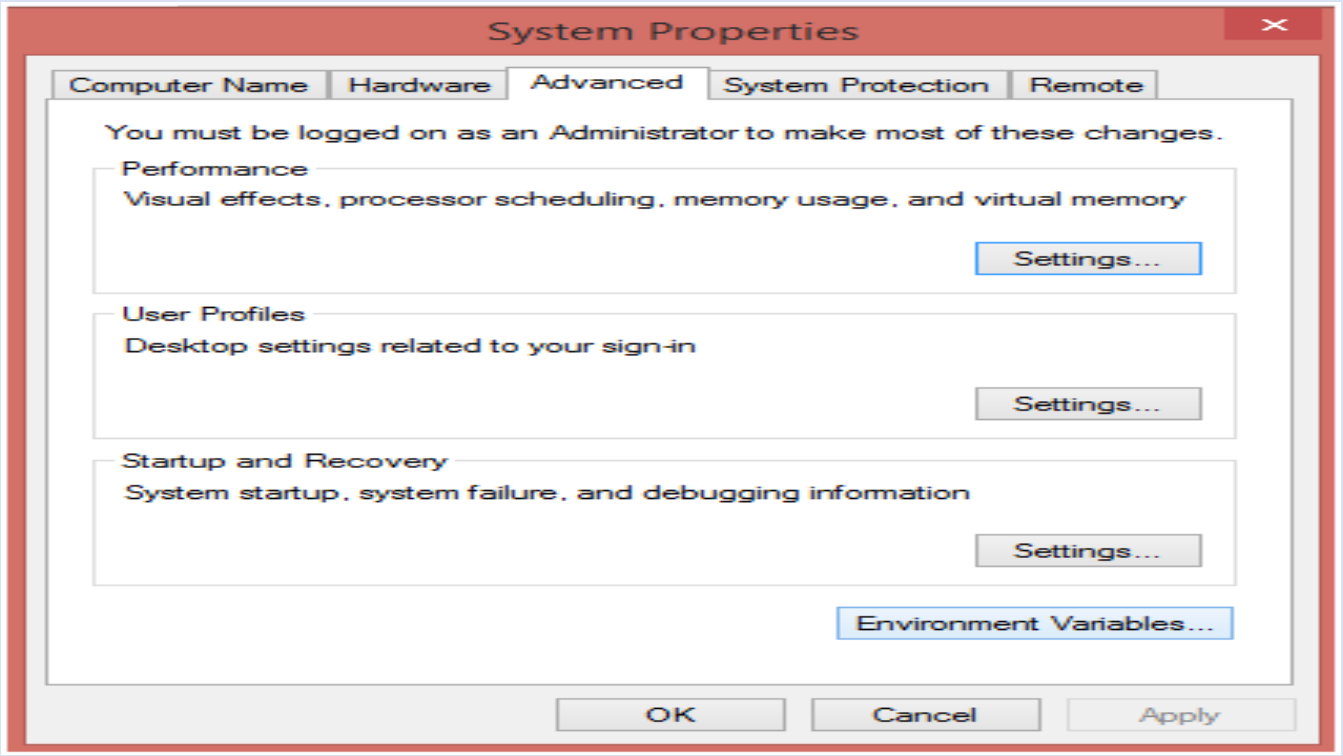
To execute Java applications from command line, we need to set Java Path. To set the path, follow the following steps.



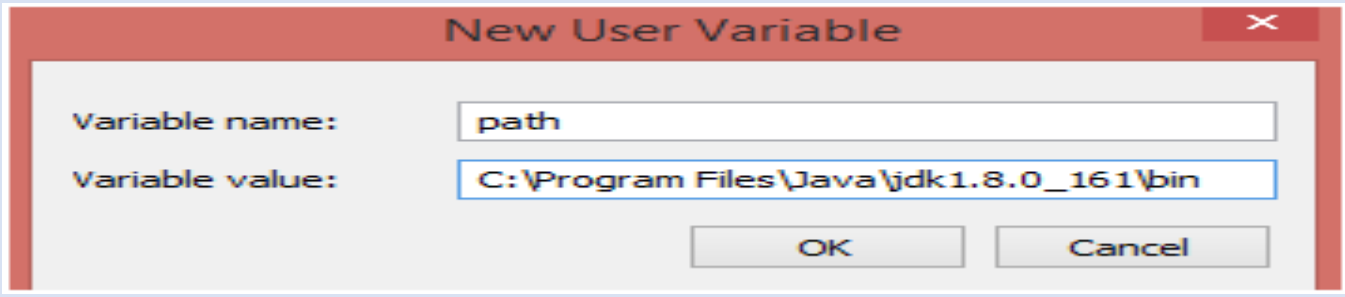
Right click on "this PC". It can be named as "My Computer" in some systems. Choose "properties" from the options.



The screen look alike the above image will open. Click on "Advanced system settings" to continue.



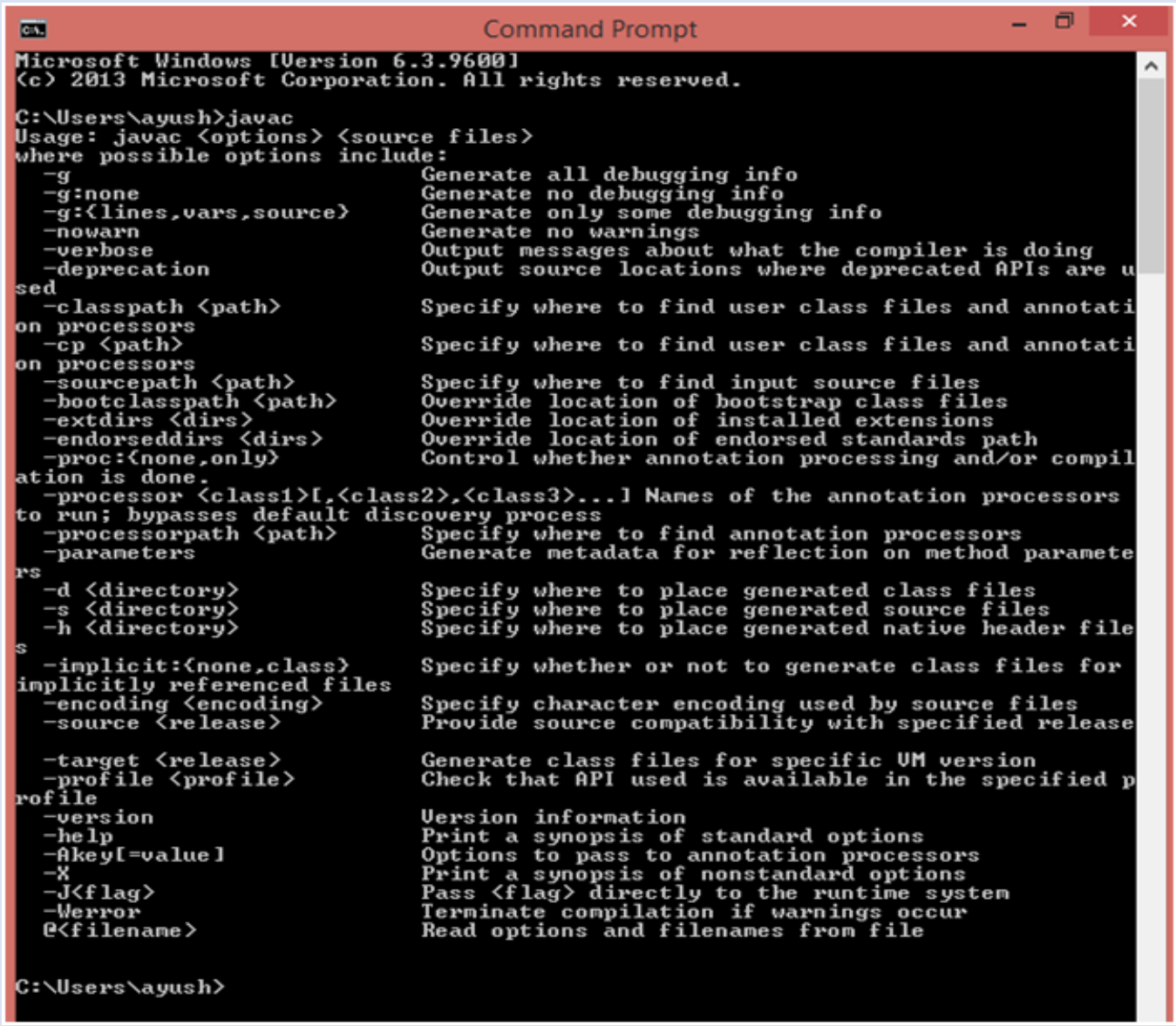
Above window will open. Click on "Environment Variables" to continue.



Enter "path" in variable name and enter the path to the bin folder inside your JDK in the variable value. Click OK.

Now Java Path has been set up. Open the Command prompt and type "javac" In case you have already open up the command prompt, I suggest you to close the existing window and reopen it again.

We will get javac executed as shown in the image below.



The Java has been installed on our system. Now, we need to configure IDEs like NetBeans or Eclipse in order to execute JavaFX applications.

What is Javac :-

javac is not Recognized

In Java, we usually get the errors and exceptions during the compilation time. But the error javac is not recognized is the most common error that is faced by many new Java programmers. In this section, we will detect why the javac command is not recognized by the compiler and what is the reasons to generate javac is not recognized. We will also see the possible solutions to resolve the error.

Before moving ahead in this section, let's understand the meaning of the error statement javac is not recognized as internal or external command.

In Java, javac is a command that compiles the Java source code. When we type the command in the Command Prompt, the prompt refuse to identify the javac command. It means that the javac.exe file is not found by the compiler. The javac.exe file exists in the bin folder of the JDK installation folder. The error we get because the PATH is not properly set. The following image shows, what the error shows.

Javac is not recognized is an error occurs while we compile the Java application. It is because the JVM is unable to find the javac.exe file. The javac.exe file is located in the bin folder of the JDK.

Syntax :-

D:\MCA-I Year>javac FileName/ClassName.java

Example :-

D:\MCA-I Year>javac Simple.java

What is Javap :-

The javap command disassembles a class file. The javap command displays information about the fields,constructors and methods present in a class file.

Syntax :-

javap Fully_Class_Name

Example :-

javap java.lang.Object

What is Javadoc :-

Creating API Document | javadoc tool

We can create document api in java by the help of javadoc tool. In the java file, we must use the documentation comment /**... */ to post information for the class, method, constructor, fields etc.

Let's see the simple class that contains documentation comment.

To create the document API, you need to use the javadoc tool followed by java file name. There is no need to compile the javafile.

On the command prompt, you need to write :- **javadoc M.java**

To generate the document api. Now, there will be created a lot of html files. Open the index.html file to get the information about the classes.

Que 02 :- Design the application by using array.

```
class array3
{
	public static void main(String args[])
	{
		int i,j,k;
		int a[][][]={{{1,2},{3,4}},{{11,22},{33,44}}} ;
		System.out.println("Multi Diamentional Array \n");
		for(i=0; i<2; i++)
		{
			for(j=0; j<2; j++)
			{
				for(k=0; k<2; k++)
				{
					System.out.print(a[i][j][k]+"\t");
				}
				System.out.println();
			}
			System.out.println();
		}
	}
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac array3.java
```

D:\MCA-I Year\Java Programming>java array3

Multi Diamentional Array

1 2

3 4

11 22

33 44

*/

Que 03 :- Implementation of Package, Interface and abstract class.

Package Program :-

```
package Student;

class PackageExample

{

	int Id;

	String Name;

	String Address;

	public static void main(String args[])

	{

		PackageExample PE = new PackageExample();

		PE.set(1,"Akshay", "Atpadi");

		PE.get();

	}

	void set(int i, String n, String a)

	{

		Id=i;

		Name=n;

		Address=a;

	}

	void get()

	{

		System.out.println("Student Id = "+Id);
```

```
            System.out.println("Student Name = "+Name);
            System.out.println("Student Address = "+Address);
        }
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac -d . PackageExample.java
D:\MCA-I Year\Java Programming>java Student.PackageExample
Student Id = 1
Student Name = Akshay
Student Address = Atpadi
*/
```

Interface Program :-

```
interface MSG
{
        public void Show1();
        public void Show2();
        public void Show3();
}
class InterfaceExample implements MSG
{
        public void Show1()
        {
```

```
        System.out.println("Wel-Come Show1 Method");
    }
    public void Show2()
    {
        System.out.println("Wel-Come Show2 Method");
    }
    public void Show3()
    {
        System.out.println("Wel-Come Show3 Method");
    }
    public static void main(String args[])
    {
        MSG obj = new InterfaceExample();
        obj.Show1();
        obj.Show2();
        obj.Show3();
    }
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac InterfaceExample.java
D:\MCA-I Year\Java Programming>java InterfaceExample
Wel-Come Show1 Method
Wel-Come Show2 Method
```

```
Wel-Come Show3 Method
*/
```

Abstract Class Program :-

```
abstract class A
{
	abstract void callme();
	public void show()
	{
		System.out.println("this is non-abstract method");
	}
}
class AbstractClassExample extends A
{
	void callme()
	{
		System.out.println("Calling...");
	}
	public static void main(String[] args)
	{
		AbstractClassExample b = new AbstractClassExample();
		b.callme();
		b.show();
```

```
        }
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac AbstractClassExample.java
D:\MCA-I Year\Java Programming>java AbstractClassExample
Calling...
this is non-abstract method
*/
```

Que 04 :- Design application using string, string builder and string tokenizer.

String Program :-

```
public class StringExample
{
	public static void main(String args[])
	{
		String s1="Java";
		char ch[]={'S','t','r','i','n','g','s'};
		String s2=new String(ch);
		String s3=new String("Example");
		System.out.println(s1);
		System.out.println(s2);
		System.out.println(s3);
	}
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac StringExample.java
D:\MCA-I Year\Java Programming>java StringExample
Java
Strings
Example
*/
```

String Builder Program :-

```
class AllStringBuilderExample
{
	public static void main(String args[])
	{
		StringBuilder sb1=new StringBuilder("Hello");
		StringBuilder sb2=new StringBuilder("Hello");
		StringBuilder sb3=new StringBuilder("Hello");
		StringBuilder sb4=new StringBuilder("Hello");
		StringBuilder sb5=new StringBuilder("Hello");
		StringBuilder sb6=new StringBuilder("Hello");
		StringBuilder sb7=new StringBuilder("Hello");
		sb1.append("Java");
		System.out.println(sb1);
		sb2.insert(1,"Java");
		System.out.println(sb2);
		sb3.replace(1,3,"Java");
		System.out.println(sb3);
		sb4.delete(1,3);
		System.out.println(sb4);
		sb5.reverse();
		System.out.println(sb5);
		sb6.append("Hello");
```

```
            System.out.println(sb6.capacity());
            System.out.println(sb6);
            sb7.ensureCapacity(50);
            System.out.println(sb7.capacity());
            System.out.println(sb7);
        }
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac AllStringBuilderExample.java
D:\MCA-I Year\Java Programming>java AllStringBuilderExample
HelloJava
HJavaello
HJavalo
Hlo
olleH
21
HelloHello
50
Hello
*/
```

String Tokenizer Program :-

```
import java.util.StringTokenizer;
```

```
public class StringTokenizerExample
{
	public static void main(String args[])
	{
		StringTokenizer st = new StringTokenizer("My Name is Akshay"," ");
		while (st.hasMoreTokens())
		{
			System.out.println(st.nextToken());
		}
	}
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac StringTokenizerExample.java
D:\MCA-I Year\Java Programming>java StringTokenizerExample
My
Name
is
Akshay
*/
import java.util.*;
public class StringTokenizerExample1
{
```

```
    public static void main(String[] args)
    {
        StringTokenizer st = new StringTokenizer("my,name,is,khan");
        System.out.println("Next token is : " + st.nextToken(","));
    }
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac StringTokenizerExample1.java
D:\MCA-I Year\Java Programming>java StringTokenizerExample1
Next token is : my
*/
```

Que 05 :- Test any five of standard exceptions and user defined custom exceptions in java.

Exception Class :-

```
public class InsufficientAmountException extends Exception
{
	InsufficientAmountException(String s)
	{
		super(s);
	}
}
```

Program :-

```
import java.util.Scanner;
public class Account
{
	static void Withdraw(int Amount) throws
InsufficientAmountException
	{
		if(Amount>20000)
		{
			throw new InsufficientAmountException("Insufficient
			Account Balance to Withdraw Money");
		}
		else
		{
```

```
                System.out.println("Please Withdraw Money");
            }
        }
        public static void main(String args[])
        {
            try
            {
                Scanner sc = new Scanner(System.in);
                int Amount1;
                System.out.println("Enter the Amount");
                Amount1=sc.nextInt();
                Withdraw(Amount1);
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
            System.out.println("Thanks For Visiting");
        }
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac Account.java
D:\MCA-I Year\Java Programming>java Account
```

Enter the Amount

10000

Please Withdraw Money

Thanks For Visiting

D:\MCA-I Year\Java Programming>javac Account.java

D:\MCA-I Year\Java Programming>java Account

Enter the Amount

25000

InsufficientAmountException: Insufficient Account Balance to Withdraw Money

Thanks For Visiting

*/

Que 06 :- Threads creation and design application by using extending the thread class/Implementing the Runnable Interface. Application of multithreading in java.

```
class RunnableDemo implements Runnable
{
        private Thread t;
        private String threadName;
        RunnableDemo( String name)
        {
                threadName = name;
                System.out.println("Creating " +  threadName );
        }
        public void run()
        {
                System.out.println("Running " +  threadName );
                try
                {
                        for(int i = 2; i > 0; i--)
                        {
                                System.out.println("Thread:" + threadName + ", " + i);
                                Thread.sleep(50);
                        }
                }
```

```
            catch (InterruptedException e)
            {
                System.out.println("Thread " +  threadName + "
                interrupted.");
            }
            System.out.println("Thread " +  threadName + " exiting.");
        }
        public void start ()
        {
            System.out.println("Starting " +  threadName );
            if (t == null)
            {
                t = new Thread (this, threadName);
                t.start ();
            }
        }
}
public class MultiThread
{
        public static void main(String args[])
        {
            RunnableDemo R1 = new RunnableDemo( "Thread-1");
            R1.start();
```

```
            RunnableDemo R2 = new RunnableDemo( "Thread-2");
            R2.start();
        }
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac MultiThread.java
D:\MCA-I Year\Java Programming>java MultiThread
Creating Thread-1
Starting Thread-1
Creating Thread-2
Starting Thread-2
Running Thread-1
Running Thread-2
Thread: Thread-2, 2
Thread: Thread-1, 2
Thread: Thread-2, 1
Thread: Thread-1, 1
Thread Thread-2 exiting.
Thread Thread-1 exiting.
*/

class ThreadDemo extends Thread
{
```

```
private Thread t;
private String threadName;
ThreadDemo( String name)
{
        threadName = name;
        System.out.println("Creating " +  threadName );
}
public void run()
{
        System.out.println("Running " +  threadName );
        try
        {
                for(int i = 2; i > 0; i--)
                {
                        System.out.println("Thread:" + threadName + ", " + i);
                        Thread.sleep(50);
                }
        }
        catch (InterruptedException e)
        {
                System.out.println("Thread " +  threadName + "
                interrupted.");
        }
```

```
            System.out.println("Thread " +  threadName + " exiting.");
        }
        public void start ()
        {
            System.out.println("Starting " +  threadName );
            if (t == null)
            {
                t = new Thread (this, threadName);
                t.start ();
            }
        }
}
public class MultiThread1
{
        public static void main(String args[])
        {
            ThreadDemo T1 = new ThreadDemo( "Thread-1");
            T1.start();
            ThreadDemo T2 = new ThreadDemo( "Thread-2");
            T2.start();
        }
}
/* OUTOUT
```

D:\MCA-I Year\Java Programming>javac MultiThread1.java

D:\MCA-I Year\Java Programming>java MultiThread1

Creating Thread-1

Starting Thread-1

Creating Thread-2

Starting Thread-2

Running Thread-1

Running Thread-2

Thread: Thread-2, 2

Thread: Thread-1, 2

Thread: Thread-2, 1

Thread: Thread-1, 1

Thread Thread-2 exiting.

Thread Thread-1 exiting.

*/

Que 07 :- Design java application using collection in java such as array list , Link list.

Array List Program :-

```
import java.util.*;

public class ArrayListExample1

{

	public static void main(String args[])

	{

		ArrayList<String> list=new ArrayList<String>();

		list.add("Mango");

		list.add("Apple");

		list.add("Banana");

		list.add("Grapes");

		System.out.println(list);

	}

}

/* OUTPUT

D:\MCA-I Year\Java Programming>javac ArrayListExample1.java

D:\MCA-I Year\Java Programming>java ArrayListExample1

[Mango, Apple, Banana, Grapes]

*/
```

Link List Program :-

```
import java.util.*;
public class LinkListExample
{
	public static void main(String args[])
	{
		LinkedList  l1= new LinkedList();
		l1.add("Mango");
		l1.add("Apple");
		l1.add("Banana");
		l1.add("Grapes");
		System.out.println("Element are "+l1);
		l1.remove(3);
		System.out.println("Element are"+l1);
	}
}
/* OUTPUT
D:\MCA-I Year\Java Programming>javac LinkListExample.java
Note: LinkListExample.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
D:\MCA-I Year\Java Programming>java LinkListExample
Element are [Mango, Apple, Banana, Grapes]
Element are[Mango, Apple, Banana]
*/
```

Que 08 :- Design GUI based java application using AWT, Swing with Event Handling.

```
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.BorderLayout;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
public class GUIApplication implements ActionListener {
	private int Count = 0;
	private JLabel lebal;
	private JFrame frame;
	private JPanel panel;
	public GUIApplication()
	{
		frame = new JFrame();
		JButton button = new JButton("Click Me");
		button.addActionListener(this);
		lebal = new JLabel("Number of Clicks : 0");
		JPanel panel = new JPanel();
		panel.setBorder(BorderFactory.createEmptyBorder(30, 30, 10,
30));
		panel.setLayout(new GridLayout(0, 1));
		panel.add(button);
		panel.add(lebal);
		frame.add(panel, BorderLayout.CENTER);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
            frame.setTitle("Your GUI Application");
            frame.pack();
            frame.setVisible(true);
        }
        public void actionPerformed(ActionEvent e)
        {
            Count++;
            lebal.setText("Number of Clicks :" +Count);
        }
        public static void main(String[] args) {
            // TODO Auto-generated method stub
            new GUIApplication();
        }
}
```


Your GUI Application
Click Me
Number of Clicks :3

Que 09 :- Design and Implement JDBC Applications.

```
package JDBCExample;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public final class JDBCExample extends javax.swing.JFrame {

    public JDBCExample() {

        initComponents();

        connect();

        load();

    }

    Connection con;

    PreparedStatement pat;

    DefaultTableModel dr;

    public void load()
```

```
    {
        int a;
        try {
            pat=con.prepareStatement("select * from jdbcapplication");
            ResultSet rs=pat.executeQuery();
            ResultSetMetaData rd=rs.getMetaData();
            a=rd.getColumnCount();
            dr=(DefaultTableModel)jTable1.getModel();
            dr.setRowCount(0);
            while(rs.next())
            {
                Vector v2=new Vector();
                for (int i = 1; i <= a; i++) {
                    v2.add(rs.getString("ID"));
                    v2.add(rs.getString("Name"));
                    v2.add(rs.getString("Mobile"));
                    v2.add(rs.getString("Address"));
                }
                dr.addRow(v2);
            }
        } catch (SQLException ex) {
Logger.getLogger(JDBCExample.class.getName()).log(Level.SEVERE, null,
ex);
```

```
        }
    }
    public void connect()
    {
        try {
            Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection("jdbc:mysql://localhost/librarydatabase",
"root","");
        } catch (ClassNotFoundException | SQLException ex) {
     Logger.getLogger(JDBCExample.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
    private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
        try {
                String Name=txtName.getText();
                String Mobile=txtMobile.getText();
                String Address=txtAddress.getText();
                pat=con.prepareStatement("insert into
jdbcapplication(Name,Mobile,Address)values(?,?,?)");
                pat.setString(1,Name);
                pat.setString(2,Mobile);
                pat.setString(3,Address);
                pat.executeUpdate();
```

```
            JOptionPane.showMessageDialog(this, "Information Added");
            txtName.setText("");
            txtMobile.setText("");
            txtAddress.setText("");
            txtName.requestFocus();
            load();
        } catch (SQLException ex) {
Logger.getLogger(JDBCExample.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
   private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
        dr=(DefaultTableModel)jTable1.getModel();
        int selected=jTable1.getSelectedRow();
        int id=Integer.parseInt(dr.getValueAt(selected, 0).toString());
        String Name=txtName.getText();
        String Mobile=txtMobile.getText();
        String Address=txtAddress.getText();
        try {
            pat=con.prepareStatement("update jdbcapplication set Name
=?,Mobile =?,Address =? where id =?");
            pat.setString(1,Name);
            pat.setString(2,Mobile);
            pat.setString(3,Address);
```

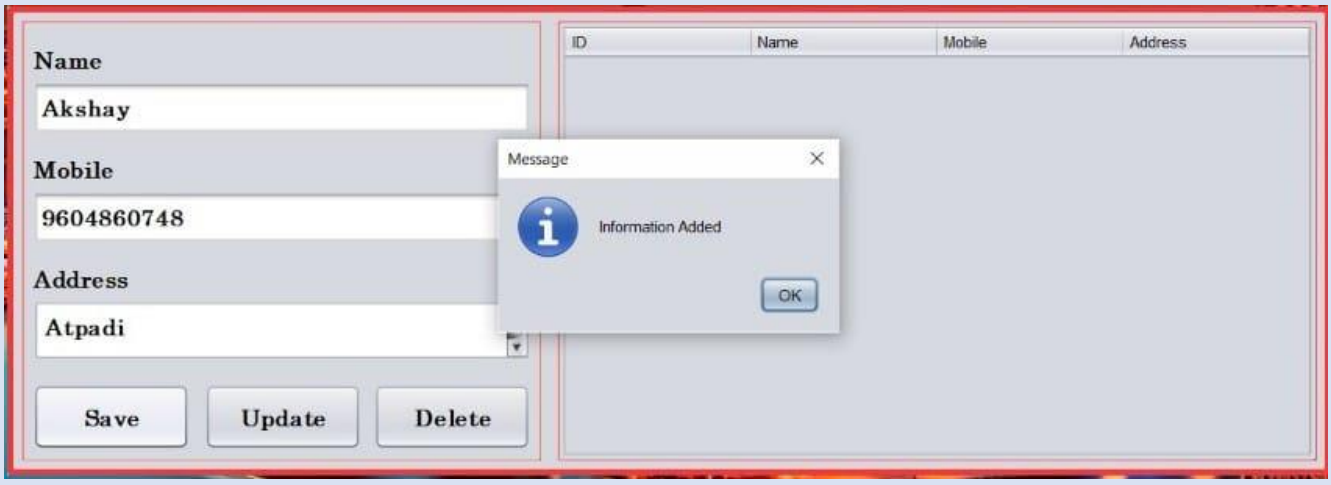
```
            pat.setInt(4,id);
            pat.executeUpdate();
            JOptionPane.showMessageDialog(this, "Information Updated");
            txtName.setText("");
            txtMobile.setText("");
            txtAddress.setText("");
            txtName.requestFocus();
            load();
            btnSave.setEnabled(true);
        } catch (SQLException ex) {

Logger.getLogger(JDBCExample.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
        dr=(DefaultTableModel)jTable1.getModel();
        int selected=jTable1.getSelectedRow();
        int id=Integer.parseInt(dr.getValueAt(selected, 0).toString());
        try {
                pat=con.prepareStatement("delete from jdbcapplication where id
=?");
                pat.setInt(1,id);
                pat.executeUpdate();
```

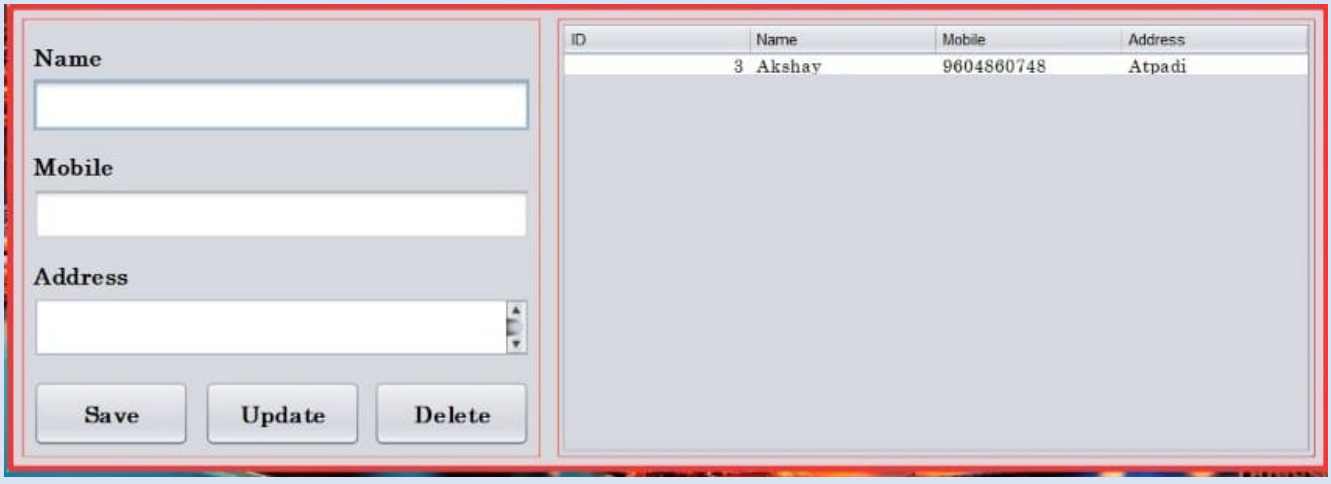
```
            JOptionPane.showMessageDialog(this, "Information Deleted");
            txtName.setText("");
            txtMobile.setText("");
            txtAddress.setText("");
            txtName.requestFocus();
            load();
            btnSave.setEnabled(true);
        } catch (SQLException ex) {
     Logger.getLogger(JDBCExample.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        dr=(DefaultTableModel)jTable1.getModel();
        int selected=jTable1.getSelectedRow();
        int id=Integer.parseInt(dr.getValueAt(selected, 0).toString());
        txtName.setText(dr.getValueAt(selected, 1).toString());
        txtMobile.setText(dr.getValueAt(selected, 2).toString());
        txtAddress.setText(dr.getValueAt(selected, 3).toString());
        btnSave.setEnabled(false);
    }

    public static void main(String args[]) {
```

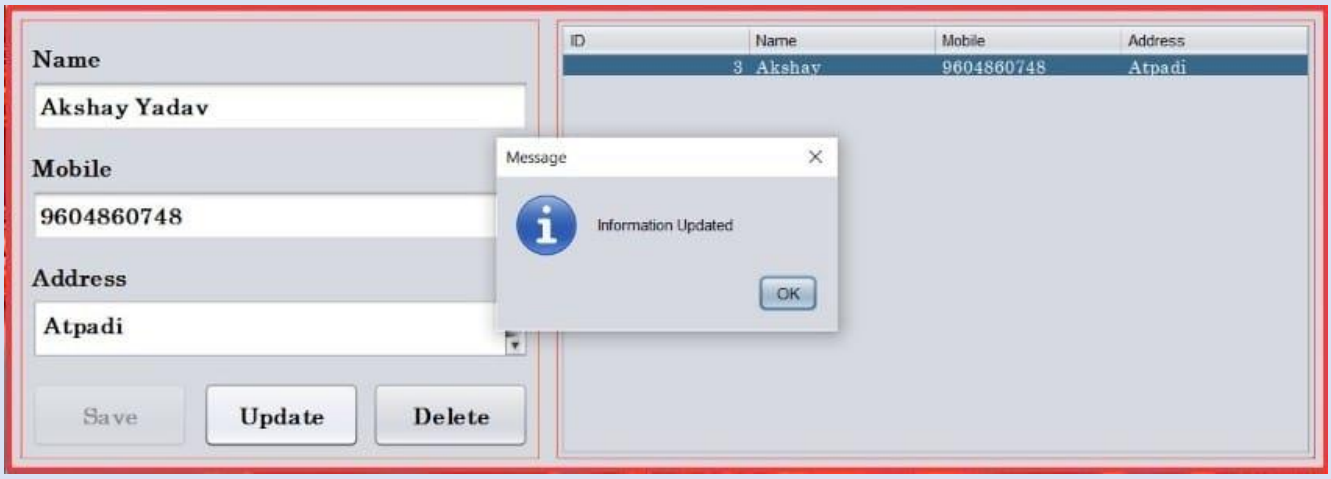
```
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
java.util.logging.Logger.getLogger(JDBCExample.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(JDBCExample.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(JDBCExample.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(JDBCExample.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new JDBCExample().setVisible(true);
            }
```

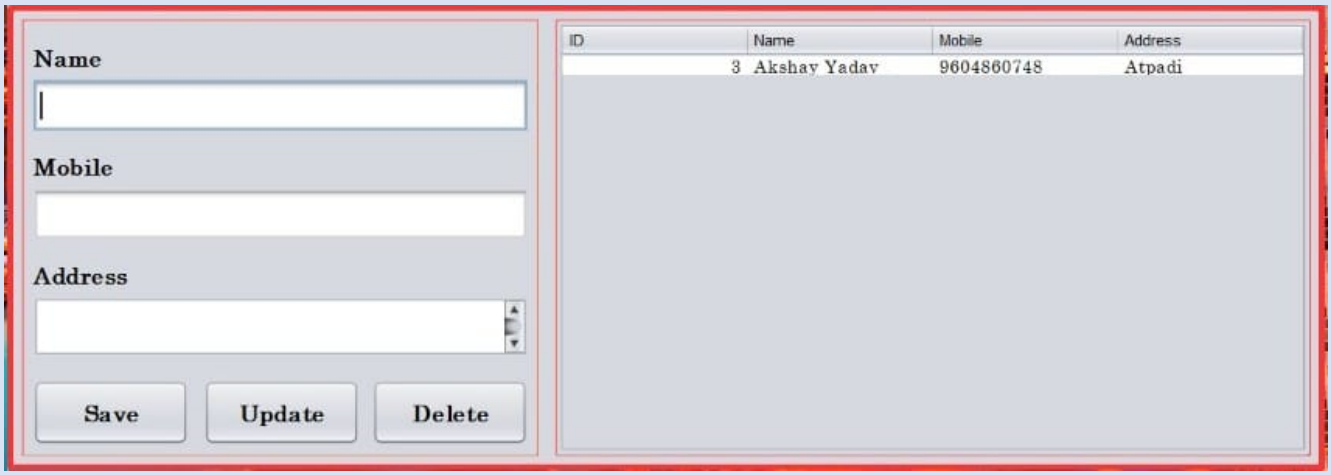
```
        });

    }

}
```

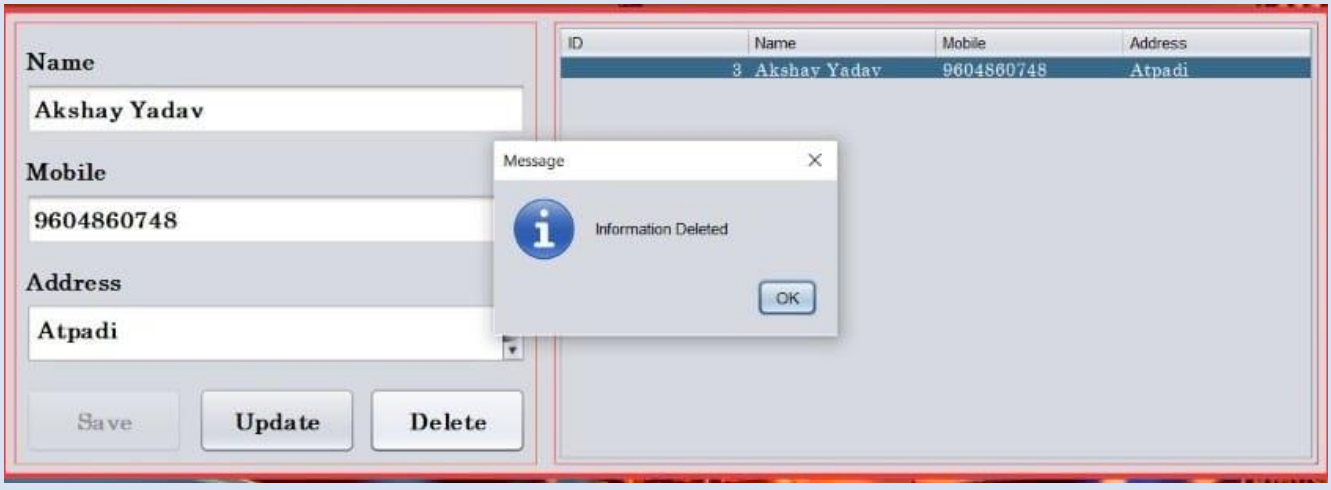

Name
Mobile
Address
Save
Update
Delete
ID
Name
Mobile
Address


Name
Akshay
Mobile
9604860748
Address
Atpadi
Save
Update
Delete
ID
Name
Mobile
Address
Message
Information Added
OK


Name
Mobile
Address
Save
Update
Delete
ID
Name
Mobile
Address
3
Akshay
9604860748
Atpadi


Name
Akshay Yadav
Mobile
9604860748
Address
Atpadi
Save
Update
Delete
ID
Name
Mobile
Address
3 Akshay 9604860748 Atpadi
Message
Information Updated
OK


Name
Mobile
Address
Save
Update
Delete
ID
Name
Mobile
Address
3 Akshay Yadav 9604860748 Atpadi


Name
Akshay Yadav
Mobile
9604860748
Address
Atpadi
Save
Update
Delete
ID
Name
Mobile
Address
3 Akshay Yadav 9604860748 Atpadi
Message
Information Deleted
OK

Que 10 :- Design and Implement servlet Application.

Que 11 :- Design and Implement JSP Application