Pimpri Chinchwad College of Engineering
Computer Laboratory - III

Assignment No - 1

1 Aim

Write a program in python/ Java/ Scala/ C++/ HTML5 to implement password data encryption. Use encryption method overloading.

2 Objective

- To encrypt data using password for preventing unauthorized access.

3 Software Requirements

- Operating System:Windows 10, Ubuntu.
- Java

4 Mathematical Model

S = s, e, x, y, Fme, DD, NDD

S = Initial State
E = End State
X = Input Value
X={x1,x2}

where, x1=UserName
x2=Password
Y =Output
Y={y1}
y1={"Successfully added into Database}
Fme = Main function
Fme={f1}
f1="MD5 Algorithm to calulate Hash of the Password".
H=MD5(x2)
where, H=Hash of the Password.
DD = Deterministic data {username,password}
NDD = Non Deterministic Data { Hash}

5 Theory

5.1 MD5

- MD5 algorithm can be used as a digital signature mechanism.The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, mostly expressed in text format as a 32-digit hexadecimal number. MD5 has been utilized in a variety of cryptographic applications and is also commonly used to verify data integrity.

5.2 Password Encryption

Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Unencrypted data is called plain text ; encrypted data is referred to as cipher text.

5.3 Method Overloading

Method Overloading is a feature that allows a class to have two or more methods having same name, if their argument lists are different.

```
class CalHash{

     String hash(String pass)
     {
     String hashcode=getHash(pass);
     return hashcode;
     }

     String hash(String pass,StringSalt)
     {
     String hashcode = getHash(pass+salt);
     return hashcode;
     }

     public static void main(String args[]){
     CalHash obj=new CalHash();

     if(pass.length()<8)
       obj.hash(pass,salt);
     else
       obj.hash(pass);

     }
   }
```

5.4 Java Swing

Swing API is set of extensible GUI Components to ease developer's life to create JAVA based Front End/ GUI Applications. It is build upon top of AWT API and acts as replacement of AWT API as it has almost every control corresponding to AWT controls. Swing component follows a Model-View-Controller architecture to fulfill the following criteria.

- A single API is to be sufficient to support multiple look and feel.
- API is to model driven so that highest level API is not required to have the data.
- API is to use the Java Bean model so that Builder Tools and IDE can provide better services to the developers to use it.

CODE:

```
package com.pratiksha.cl3.assignment1;

import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

import java.sql.*;

public class Login extends JFrame implements ActionListener

{

JLabel l1, l2, l3;

JTextField tf1;

JButton btn1;

JPasswordField p1;
JCheckBox checkbox;

Login()

{

setTitle("Login Form in Windows Form");

setVisible(true);

setSize(800, 800);

setLayout(null);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

l1 = new JLabel("Login Form in Windows Form:");

l1.setForeground(Color.blue);
```

```
l1.setFont(new Font("Serif", Font.BOLD, 20));

l2 = new JLabel("Enter Email:");

l3 = new JLabel("Enter Password:");

tf1 = new JTextField();

p1 = new JPasswordField();

checkbox = new JCheckBox("Security");

btn1 = new JButton("Submit");

l1.setBounds(100, 30, 400, 30);

l2.setBounds(80, 70, 200, 30);

l3.setBounds(80, 110, 200, 30);

tf1.setBounds(300, 70, 200, 30);

p1.setBounds(300, 110, 200, 30);

checkbox.setBounds(300, 160, 100, 100);

btn1.setBounds(150, 160, 100, 30);

add(l1);

add(l2);

add(tf1);

add(l3);

add(p1);

add(checkbox);
```

```
add(btn1);

btn1.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

showData();

}

public void showData()

{

JFrame f1 = new JFrame();

JLabel l, l0;

String str1 = tf1.getText();

char[] p = p1.getPassword();

String str2 = new String(p);

try

{

// set state
checkbox.setSelected(true);

// check state

Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
PreparedStatement ps;
if (checkbox.isSelected()) {
```

```
ps = con.prepareStatement("select name from regform where user=? and passO=?");
ps.setString(1, str1);
String password = MD5Calculation.encrypt(str2, "MD5", str1);
ps.setString(2, password);
} else {

ps = con.prepareStatement("select name from regform where user=? and pass=?");
ps.setString(1, str1);
String password = MD5Calculation.encrypt(str2, "MD5");
ps.setString(2, password);
}

ResultSet rs = ps.executeQuery();

if (rs.next())

{

f1.setVisible(true);

f1.setSize(600, 600);

f1.setLayout(null);

l = new JLabel();

l0 = new JLabel("you are succefully logged in..");

l0.setForeground(Color.blue);

l0.setFont(new Font("Serif", Font.BOLD, 30));

l.setBounds(60, 50, 400, 30);

l0.setBounds(60, 100, 400, 40);

f1.add(l);

f1.add(l0);

l.setText("Welcome " + rs.getString(1));
```

```
l.setForeground(Color.red);

l.setFont(new Font("Serif", Font.BOLD, 30));

} else

{

JOptionPane.showMessageDialog(null,

"Incorrect email-Id or password..Try Again with correct detail");

}

}

catch (Exception ex)

{

System.out.println(ex);

}

}

public static void main(String arr[])

{

new Login();

}
}
```

6 Algorithms

6.1 Algorithm for Password encryption

1. Take Username and Password from user.
2. If password length is smaller than 8, Add Pepper and salt into password. Else leave as it is.
3. Calculate hash for the Password.
4. Store Hash Value into Database.
5. Use the stored hash for authenticating user.

7 Testing

7.1 Positive Testing

Input:Enter correct username and password
username=”pccoe”
password=”pccoe”
output: Successfully Login

7.2 Negative Testing

Input: Enter incorrect username and password.
username=”pccoe”
password=”1234”
output: Password incorrect

8 Conclusion

Thus, we have studied and implemented password data encryption by calculating hash value of password and stored into database.

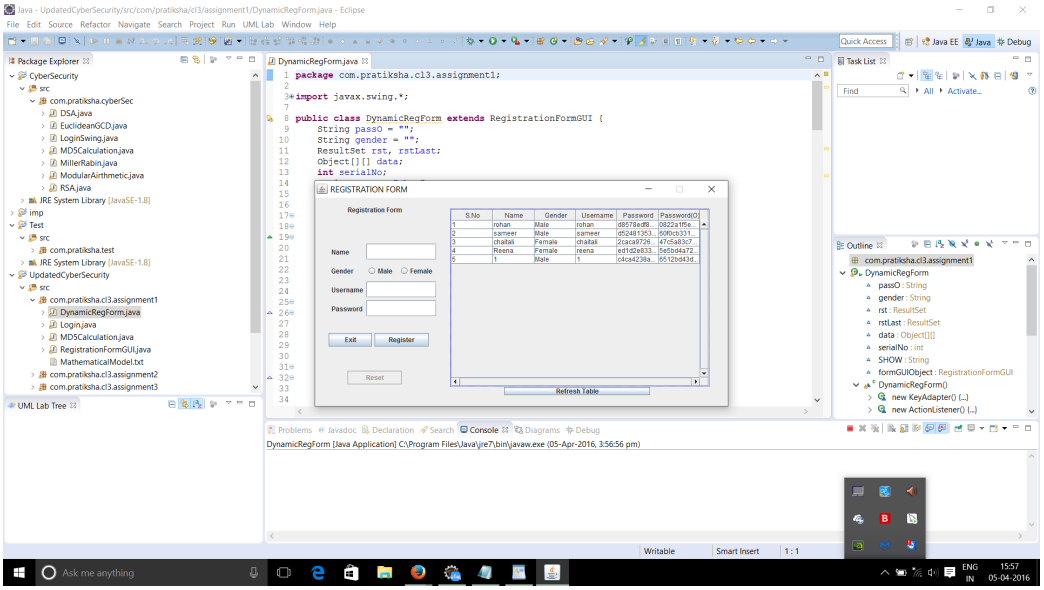
Roll No.	Name of Student	Date of Submission
302	Abhinav Bakshi	31/12/2015

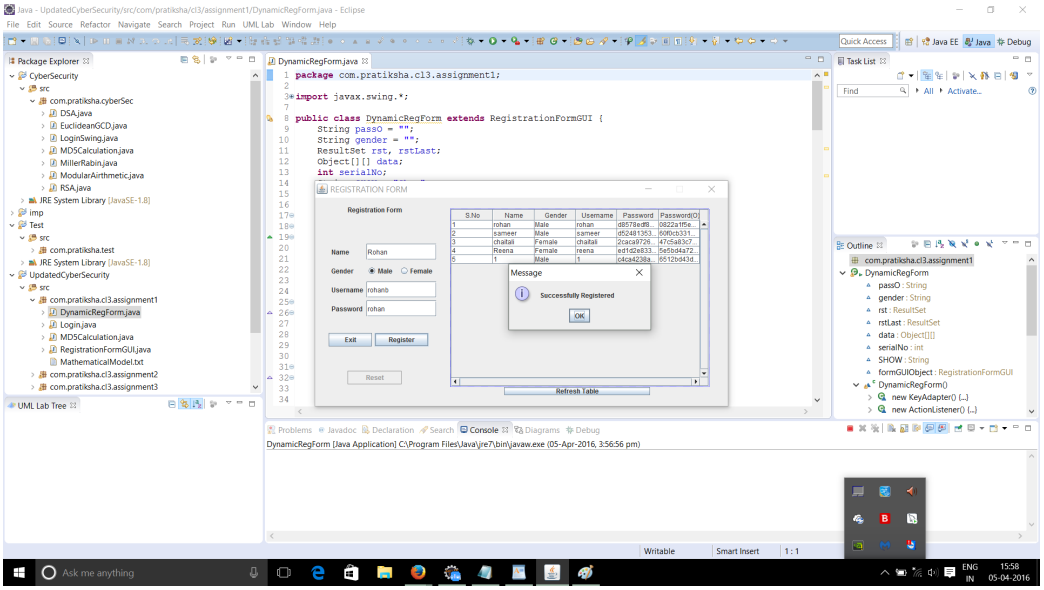
9 Plagiarism Report

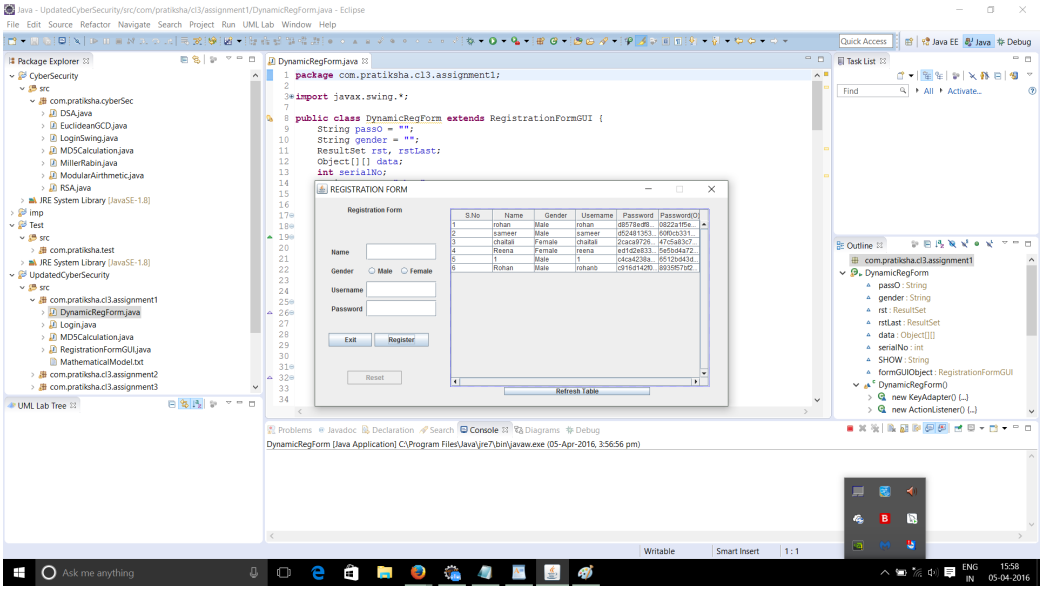
Completed: 100% Checked — 79% Unique

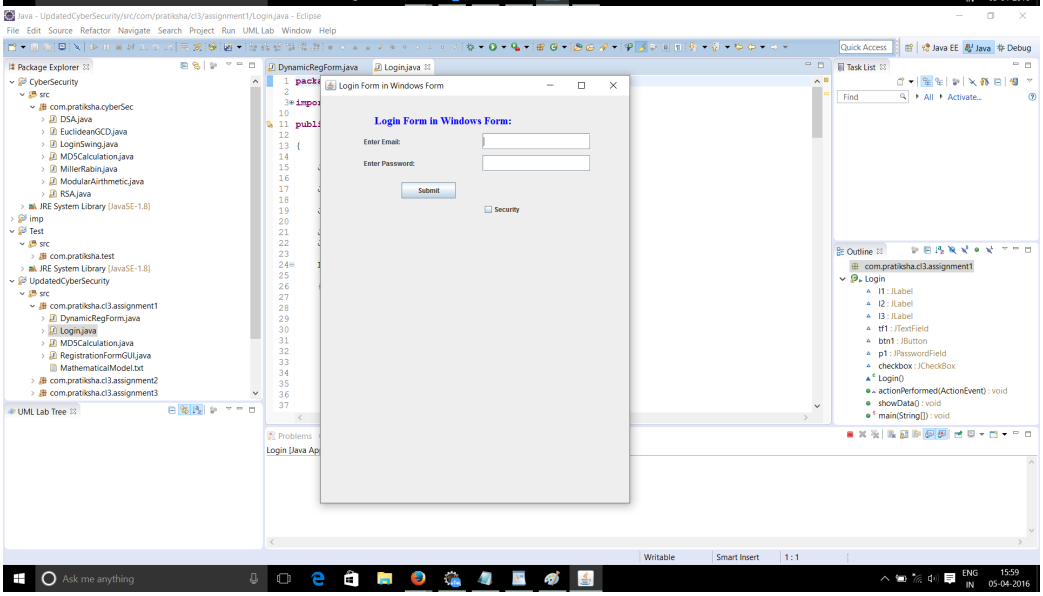
%%%% \section{Aim} \paragraph{}Write a program in python/	- Unique
Use encryption method overloading. \section{Objective} \begin{itemize}	- Unique
access. \end{itemize} \section{Software Requirements} \begin{itemize}	- Unique
\section{Mathematical Model} \paragraph{} S = {s, e, x,	- Unique
= End State \\ X = Input Value\\ X=\{x1,x2\} \\ where, x1=UserName\\	- Unique
added into Database\}\\ Fme = Main function \\ Fme=\{f1\}\\f1="MD5	- Unique
where, H=Hash of the Password.\\ DD = Deterministic data	- Unique
Hash\} \\ Memshared = \section{Theory} \subsection{MD5}	- Unique
signature mechanism.The MD5 message-digest algorithm is	- Unique
(16-byte) hash value, mostly expressed in text format as	- Unique
a variety of cryptographic applications and is also commonly	- Unique
it is neither encryption nor encoding. It cannot be reversed	- Plagiarized
that is used to verify data integrity through the creation	- Plagiarized
a message of any length) that is claimed to be as unique	- Plagiarized

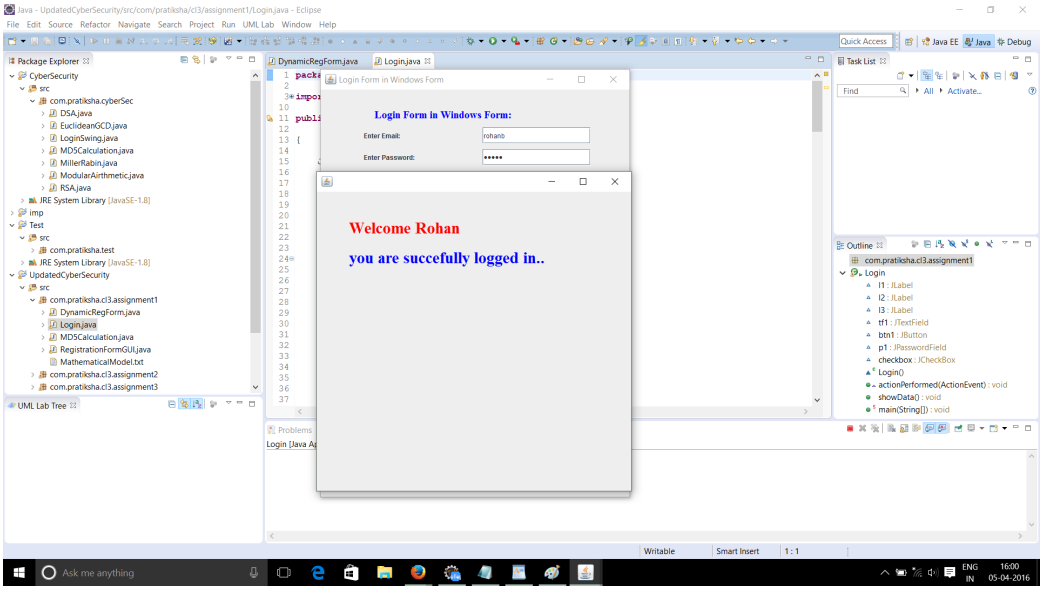
10 Output

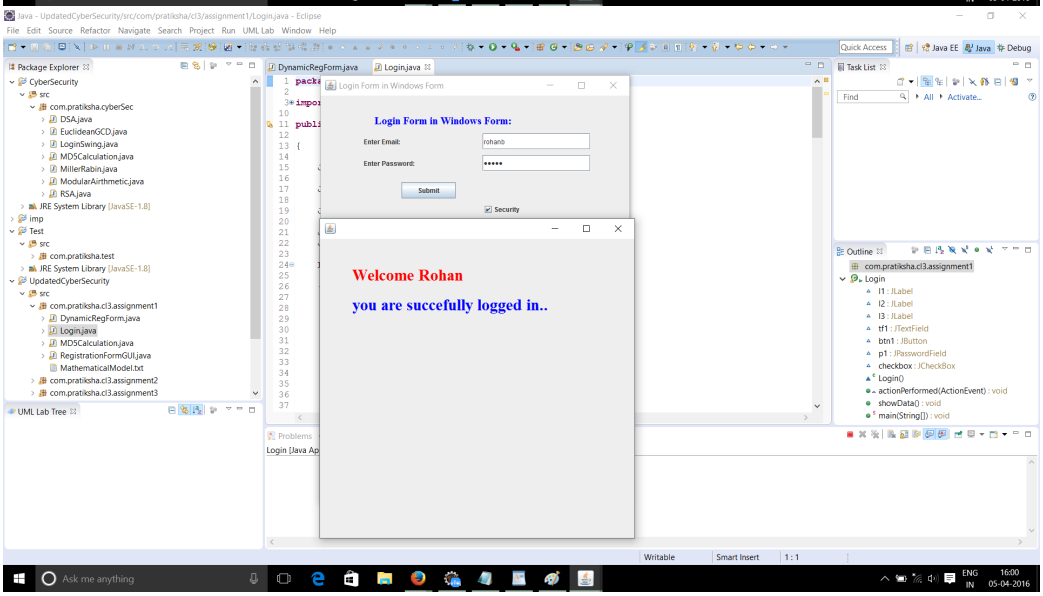





REGISTRATION FORM
Registration Form
Name
Gender
Male
Female
Username
Password
Exit
Register
Reset
S.No
Name
Gender
Username
Password
Refresh Table


Login Form in Windows Form
Login Form in Windows Form:
Enter Email:
Enter Password:
Submit
Security


Login Form in Windows Form:
Enter Email:
rohanb
Enter Password:
Welcome Rohan
you are succefully logged in..


Login Form in Windows Form:
Enter Email:
rohanb
Enter Password:
Submit
Security
Welcome Rohan
you are succefully logged in..