|  |  |
| --- | --- |
| **Ex.** 9 | **FILE HANDLING AND COMMAND LINE ARGUMENTS** |
| **Date:** 13-09-2024 | |

**PROGRAM 1**

**AIM:**

To create a Java Application to validate the username and password from a file, and upon successful validation, write command-line input into a new file.

**ALGORITHM:**

1. Accept username and password input from the user.
2. Read the stored username and password from a separate file.
3. Compare the input credentials with the stored credentials.
4. If the credentials match, allow the user to provide additional input via command-line arguments.
5. Write the contents from the command-line arguments into a new file.
6. If the credentials do not match, print an error message.
7. Use exception handling to manage file I/O operations (reading credentials and writing output).

**PROGRAM:**

package Lab9;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.util.Scanner;

import java.io.BufferedReader;

import java.io.IOException;

class DummyCredentials {

public void writingDummyCredentials(String fileName) {

File f = new File("/home/snucse/Desktop/Java/Lab9/" + fileName + ".txt");

try (FileWriter fileWriter = new FileWriter(f)) {

fileWriter.write("ashwin,ashwin2005\nprasad,prasad1971");

System.out.println("A File with Dummy Credentials is Created.");

} catch (IOException e) {

System.out.println(e);

}

}

public void readingDummyCredentials(String fileName) {

try (BufferedReader fileReader = new BufferedReader(

new FileReader(new File("/home/snucse/Desktop/Java/Lab9/" + fileName + ".txt")))) {

String line;

while ((line = fileReader.readLine()) != null) {

String credentials[] = line.split(",");

if (credentials.length == 2) {

System.out.println("User Name = " + credentials[0]);

System.out.println("Password = " + credentials[1]);

} else {

System.out.println("Invalid Format..");

}

}

} catch (Exception e) {

System.out.println(e);

}

}

public boolean credentialsChecker(String username, String password, String fileName) {

try (BufferedReader reader = new BufferedReader(

new FileReader(new File("/home/snucse/Desktop/Java/Lab9/" + fileName + ".txt")))) {

String line;

while ((line = reader.readLine()) != null) {

String content[] = line.split(",");

if (content[0].equals(username)) {

if (content[1].equals(password)) {

System.out.println("Login Credentials are Correct!");

return true;

}

}

}

} catch (Exception e) {

System.out.println(e);

}

System.out.println("Login Credentials are Incorrect..");

return false;

}

public void contentWriter(String[] args, String fileName) {

File f = new File("/home/snucse/Desktop/Java/Lab9/" + fileName + ".txt");

try (FileWriter writer = new FileWriter(f)) {

for (String con : args) {

writer.write(con + "\n");

}

} catch (Exception e) {

System.out.println(e);

}

}

public void contentReader(String fileName) {

try (BufferedReader reader = new BufferedReader(

new FileReader(new File("/home/snucse/Desktop/Java/Lab9/" + fileName + ".txt")))) {

String line;

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

} catch (Exception e) {

System.out.println(e);

}

}

}

public class ex1 {

public static void main(String[] args) {

DummyCredentials dc = new DummyCredentials();

Scanner input = new Scanner(System.in);

System.out.print("Enter the Name of the File: ");

String fileName = input.nextLine();

System.out.println("Going to Write Dummy Credentials in the File.");

dc.writingDummyCredentials(fileName);

System.out.println("Printing the Contents of the File:");

dc.readingDummyCredentials(fileName);

System.out.println("-----------------------------------------------");

System.out.print("Enter the Username: ");

String username = input.nextLine();

System.out.print("Enter the Password: ");

String password = input.nextLine();

if (dc.credentialsChecker(username, password, fileName)) {

System.out.println("-----------------------------------------------");

System.out.print("Enter the Name of the File: ");

String outputFile = input.nextLine();

System.out.println("Writing the Command Line Arguments into the File..");

dc.contentWriter(args, fileName);

System.out.println("Reading the Contents of the Output File:");

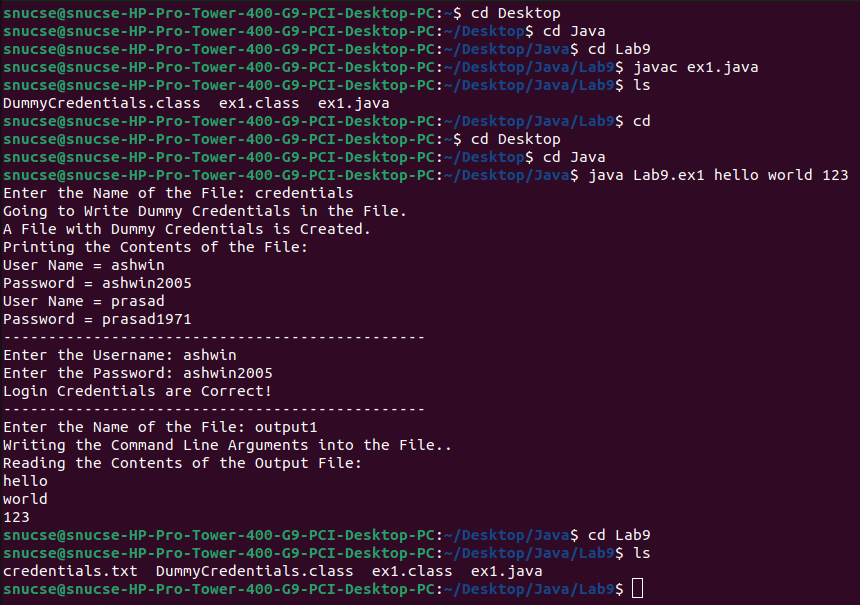
dc.contentReader(fileName);

}

}

}

**OUTPUT:**



**PROGRAM 2**

**AIM:**

To create a Java Application to create a file with 10 students and their marks, and then generate two new files: one for students scoring above 90 and another for students scoring below 40, with exception handling for file operations.

**ALGORITHM:**

1. Create a file with the names of 10 students and their marks (randomly assigned between 1 and 100).
2. Read the file and process each student's name and marks.
3. If a student has marks greater than 90, add their name to the best\_performers file.
4. If a student has marks less than 40, add their name to the low\_performers file.
5. Write the results into the respective files.
6. Use exception handling to ensure proper file reading and writing operations, catching any errors that occur during these processes.

**PROGRAM:**

package Lab9;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.BufferedReader;

import java.io.IOException;import java.util.ArrayList;

public class ex2 {

public void createBestPerformers(ArrayList<String> bestPerformers) {

try (FileWriter writer = new FileWriter(new File("/home/snucse/Desktop/Java/Lab9/best\_performers.txt"))) {

for (String con : bestPerformers) {

writer.write(con + "\n");

}

} catch (Exception e) {

System.out.println(e);

}

}

public void createLowPerformers(ArrayList<String> lowPerformers) {

try (FileWriter writer = new FileWriter(new File("/home/snucse/Desktop/Java/Lab9/low\_performers.txt"))) {

for (String con : lowPerformers) {

writer.write(con + "\n");

}

} catch (Exception e) {

System.out.println(e);

}

}

public static void main(String[] args) {

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

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

ex2 object = new ex2();

try (BufferedReader reader = new BufferedReader(

new FileReader(new File("/home/snucse/Desktop/Java/Lab9/marks.txt")))) {

String line;

while ((line = reader.readLine()) != null) {

String marks[] = line.split(",");

if (marks.length == 2) {

if (Integer.parseInt(marks[1]) > 90) {

bestPerformers.add(marks[0]);

} else if (Integer.parseInt(marks[1]) < 40) {

lowPerformers.add(marks[0]);

}

}

}

object.createBestPerformers(bestPerformers);

object.createLowPerformers(lowPerformers);

} catch (Exception e) {

System.out.println(e);

}

}

}

**OUTPUT:**

A computer screen shot of a program code

Description automatically generated

**RESULT:**

Thus, different Java Applications to Handle Text Files have been compiled and executed successfully.