# **Name: Abdurrahman Qureshi**

# **Roll No: 210451**

Practical No: 23N24N25

**1)** **WAP to accept a password from user and throw “Authentication Failure” exception if password is incorrect.**

**CODE:**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class PasswordAuthentication {

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String password = "helloWorld";

String userEnteredPassword = br.readLine();

try {

authenticate(userEnteredPassword);

System.out.println("Login Successful");}

catch (AuthencticationFailureException e) {

System.out.println(e.getMessage());} }

public static void authenticate(String pass) throws AuthencticationFailureException {

if(!pass.equals("helloWorld")) {

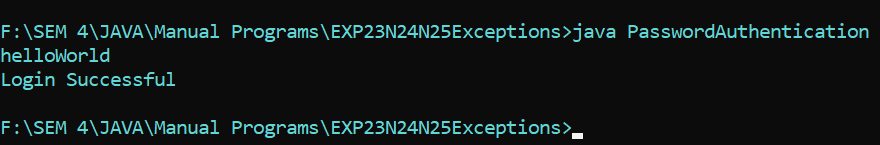
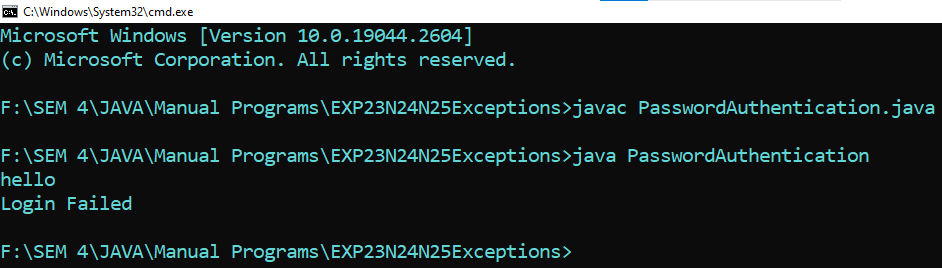
throw new AuthencticationFailureException("Login Failed");}}}

class AuthencticationFailureException extends Exception{

AuthencticationFailureException(String msg) {

super(msg);}}

**OUTPUT:**

****

**2) WAP to input name and balance of customer and throw user defined exception if balance is less than 1500**

**CODE:**

class customer {

protected String name;

protected int balance;

customer(String n , int b) {

name = n;

balance = b;

try {

CheckBalance(balance);

System.out.println("Valid Amount"); }

catch (InvalidBalance e) {

System.out.println(e.getMessage());}}

public static void CheckBalance(int bal) throws InvalidBalance {

int bal1 = bal;

if (bal1 < 1500) {

throw new InvalidBalance("Invalid balance");}}}

class InvalidBalance extends Exception {

InvalidBalance(String msg) {

super(msg);}}

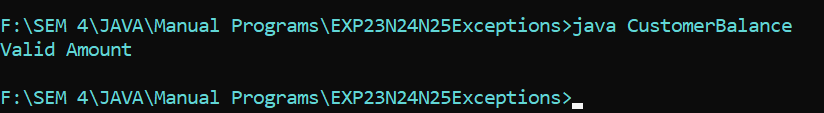
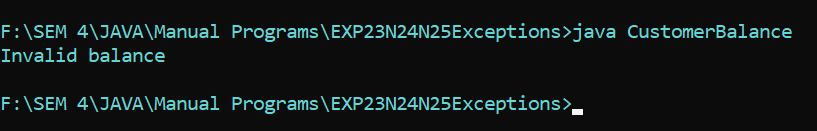
public class CustomerBalance {

public static void main(String[] args) {

customer c = new customer("Jesse" , 200);

//customer c = new customer("Jesse" , 22200);}}

**OUTPUT:**



**3)** **W** **AP input name and salary of an employee and throw user defined exception if salary is negative**

**CODE:**

class Emp {

protected String name;

protected int sal;

Emp(int s, String n) {

name = n;

sal = s;

try {

ErrorSalary(sal);

System.out.println("Valid Salary");

} catch (SalaryIsNegative e) {

System.out.println(e.getMessage());}}

static void ErrorSalary(int sala) throws SalaryIsNegative {

if (sala < 0)

throw new SalaryIsNegative("Negative Salary");}}

class SalaryIsNegative extends Exception {

SalaryIsNegative(String msg) {

super(msg);}}

public class NegativeSalary {

public static void main(String[] args) {

Emp e = new Emp(-2000, "Walter");

//Emp e = new Emp(22000, "Walter");}}

**OUTPUT:**

