**public** **class** Employee { //Employee class since data will only be accessible to employees

**private** String username;

**private** String password;

**private** **long** ID; //unique 12-digit ID number

**public** Employee() { //default constructor

**this**.username = "none";

**this**.password = "none";

**this**.ID = 000000000000;

}

**public** Employee(String username, String password, **long** ID) { //parameterized constructor

**this**.username = username;

**this**.password = password;

**this**.ID = ID;

}

**void** setUsername(String username) { //sets username

**this**.username = username;

}

String getUsername() {//returns username

**return** username;

}

**void** setPassword(String password) { //sets or resets password

**this**.password = password;

}

String getPassword() { //returns password

**return** password;

}

**void** setID(**long** ID) { //sets ID

**this**.ID = ID;

}

**long** getID() { //returns ID

**return** ID;

}

}

**================================================================================================================================================================**

**import** java.util.Scanner;

**import** java.util.InputMismatchException;

**import** java.util.LinkedList;

**public** **class** PrototypeMain {

**static** Scanner *scnr* = **new** Scanner(System.***in***); //scanner to load in information

**public** **static** **void** employeeLogin(LinkedList<Employee> employees) { //login portal

**try** {

System.***out***.println("Enter username: ");

String user = *scnr*.next(); //load username

System.***out***.println("Enter password: ");

String key = *scnr*.next(); //load password

**int** n = employees.size(); //gets current size of employee list

**int** i = 0;

**if**(n == 0) { //user cannot exist if list is empty

System.***out***.println("User does not exist");

}

**for**(i = 0; i < n; ++i) { //iterate through list to find a match to provided credentials

**if**(employees.get(i).getUsername().compareTo(user) == 0) {

//given username matches a username in the database

**if**(employees.get(i).getPassword().compareTo(key) == 0) {

//given password matches password corresponding to username in database

**try** {

System.***out***.println("Please verify identity with employee identification number: ");

**long** IDnum = *scnr*.nextLong(); //verification step

**if**(employees.get(i).getID() == IDnum) {

//ID credentials match to 12-digit ID number that corresponds to username and

//password within the database

System.***out***.println("Permission to access: Granted"); //access to data granted

**break**;

}

**else** { //ID credentials do not match to the given username and password that are

//in the database

System.***out***.println("Permission to access: Denied"); //access to data denied

**break**;

}

}

**catch**(InputMismatchException e) { //catches input if there is a mismatch to expected type

e.getMessage();

}

}

**else** { //password does not match the given username; username exists in database

System.***out***.println("Incorrect Password");

**break**;

}

}

**else** { //username does not match a username in database

System.***out***.println("User does not exist");

**break**;

}

}

}

**catch**(InputMismatchException e) { //catches input if there is a mismatch to expected type

e.getMessage();

}

}

**public** **static** **void** createAccount(LinkedList<Employee> employees) { //creates new employee account

**try** {

System.***out***.println("Enter username for new employee: ");

String user = *scnr*.next(); //accepts string for username

System.***out***.println("Enter password for new employee: ");

String key = *scnr*.next(); //accepts string for password

System.***out***.println("Assign identification number(12 digits long) for new employee: ");

**long** IDnum = *scnr*.nextLong(); //accepts long integer for ID number

Employee person = **new** Employee(user, key, IDnum); //assigns username, password, and ID to new employee

employees.add(person); //adds new employee to list of employees

}

**catch**(InputMismatchException e) { //catches input if there is a mismatch to expected type

e.getMessage();

}

}

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

LinkedList<Employee> employees = **new** LinkedList<Employee>(); //list of employees

**int** i = 0;

**try** {

**while**(i == 0) { //while loop ends after login is attempted

System.***out***.println("Sign in Page");

System.***out***.println("1. Login");

System.***out***.println("2. Create an account");

**int** option = *scnr*.nextInt();

**if**(option == 1) {

*employeeLogin*(employees); //takes user to login page

++i;

}

**if**(option == 2) {

*createAccount*(employees); //creates a new employee login

}

}

}

**catch**(InputMismatchException e) { //catches input error if input is not an integer

e.getMessage();

}

}

}