

ASSIGNMENT 3
(Due Date: 30 Nov, 2016 at 09:00 a.m.)
Total marks: 100

Objective

For this assignment, you are asked to write a Java program that demonstrate a door access system. Users are allowed to open a door if they have the access rights. There are mainly for rolls (1) Administrator, (2) Student, (3) Staff, and (4) Others. Each of users has a unique user id. Once they have logged in, the system allows them to perform corresponding actions.

Detailed Requirements

1. The program loads the door and the user data from the files “door.dat” and “user.dat” respectively, with the following format:

- Sample file of door.dat

ID	Start Time	End Time	Dept	Roll	Exception List
FSC901	08:00:00	23:00:00	SCI	ALL	none
RRS638	08:00:00	23:00:00	CS	ALL	none
AAB401	09:00:00	23:00:00	ALL	ALL	none
OEE501	09:00:00	22:30:00	SCI;BIO;C HEM;MAT H;CS	ALL	none
T1107	09:00:00	22:30:00	BIO	STAFF	none
T1109	09:00:00	22:30:00	BIO	ALL	1237;1238
T1012	09:00:00	22:30:00	BIO	STAFF	none
FSC1101	00:00:00	23:59:59	MATH	STAFF	none
FSC1102	08:00:00	22:30:00	MATH	ALL	none
T1214	00:00:00	23:59:59	CHEM	STAFF	none
HSHC3	07:00:00	19:00:00	ALL	STAFF	none
RRS204	00:00:00	23:59:59	SCI;BIO;C HEM;MAT H;CS	STUDENT	none
AAB601	00:00:00	23:59:59	ALL	STUDENT	none

- sample file of user.dat

Roll	Name	ID	Dept
Administrator	Lee Ming	admin	
Student	Chan Ming	1234	SCI
Student	Steven Hoi	1235	BIO
Student	Ho Ming	1236	CS;MATH
Student	Siu Ming	1237	CHEM
Student	Tai Ming	1238	MATH
Student	Ng Ming	1239	CS
Staff	Hui Sham	samhui	CS
Staff	Chan Tai Man	taiman	BIO
Others	Mesut Ozil	272	SPORTS
Others	Cristiano Ronaldo	cr7	SPORTS
Others	Eason Chan	eason	MUSIC

Please follow the format of “department.dat”, “door.dat” and “user.dat” when reading the files, and your program will be tested with another set of data files with the same format. Hence, you are recommended to test your program with additional input files besides the sample files.

2. Your program should read the list department from “department.dat”, as well as the door information from the “door.dat” and the user information from the “user.dat”. Your program should **recursively prompt** the user for:

```

Welcome to door access system.
Please enter a user id...or type -1 to exit
135
Invalid input
Please enter a user id...or type -1 to exit
1239
Hello, CS Student: Ng Ming
Menu:
    1   Attempt to open a door
    2   Query door opening time
    3   Switch user
   -1   Quit

```

until the user enters a valid user id or type -1 to exit.

3. When the inputted id is not an Administrator id, there are 4 options for the user. When the user choose...

Option (1):

The program should **recursively prompt** the user for a valid door id. Once a valid door id is inputted, the program should **check for access rights** and **compare the current time with the door opening time**. If a user is allowed

to open a door, the door should **record “access by who” and “access at what time”**.

Option (2):

The program should **recursively prompt** the user for a valid door id. Once a valid door id is inputted, the program should **show the opening time for this door** and **ask if the user want to open the door**.

Option (3):

The program should **recursively prompt** the user for a valid user id. Once a valid user id is inputted, the program should **greet and show menu for this user**.

Option (-1):

The program should **be terminated**.

4. When the inputted id is an Administrator id, there are 3 options for the user. When the user choose...

```
Welcome to door access system.
Please enter a user id...or type -1 to exit
admin
Hello, Administrator: Lee Ming
Menu:
    1   View a door
    2   View a user
   -1   Quit
```

Option (1):

The program should **recursively prompt** the user for a valid door id. Once a valid door id is inputted, the program should **show record of “door accessed by who” and “door accessed at what time”** if record exists.

Option (2):

The program should **recursively prompt** the user for a valid user id. Once a valid user id is inputted, the program should list out the doors that can be accessed by the user.

5. About the access rights...

Door:

- Each door can belong to one or more than one departments
- Only specify user rolls or users in exceptional list can access the door

User:

- Each user can belong to one or more than one departments
- Each user has a single roll, Administrator/Student/Staff/Others

- You are highly recommended to make use of **Inheritance and Polymorphism** in java

For Example:

T1109	09:00:00	22:30:00	BIO	ALL	1237;1238
-------	----------	----------	-----	-----	-----------

- This door can be accessed by all bio department members
- This door can also be accessed by user id:1237 and 1238

Hints

- You may store the read data into any appropriate data structure. You may also consider using ArrayList, make good use of string split.
- There are different ways to read and write files in Java. You may consider using the Scanner and PrintWriter classes.

Programming Style and Documentation

Your program must be properly indented. You should at least use the *Auto-formatting* function in *BlueJ* to help you to do proper indentation.

Your program is expected to be decorated with *internal documentation*. You need at least do the following two types of internal documentation:

Header Block

For your class file, there must be a *header* at the beginning of the file, with (1) a *short description explaining* the details and the design of your program; (2) your own name; and (3) your *UID*.

Method Header Block

For each method, there should be a header with (1) a *brief description* about what the method does and how it is achieved; (2) a brief note explaining the parameters of the method; and (3) a brief note explaining the return value of the method.

In addition to the above, wherever necessary and appropriate, you should add *inline comments* to explain the execution flow of your program. Blank lines, spaces between operators/variables (wherever appropriate) and meaningful variable names would certainly improve readability. Please bear in mind that good programming styles (indentation, comments...) are always essential.

Submission

The whole package of your program.

Marking Scheme

There are two elements in the marking scheme:

- **80%** – a working program that functions as specified
- **20%** – Programming style and documentation

Please note that submitting a program that cannot be compiled would result in very low mark.