# COMP151 Spring’17 - INIT Project

Think about a dormitory that has a secure keyless entry system. Inside its entrance hall, there is an entry system (**case sensitive**) where the dorm residents must enter their **name**, **room number,** and **password**. Upon entry of valid data, the system will unlock the inner door that leads to the dorm's living quarters.

Your task is to implement a sentry program that asks for three pieces of information: resident's name, room number, and a password. A password is any sequence of characters that are **at least** three characters long. If everything matches, then the system unlocks the door so the resident can enter. As soon as the validated resident enters, the door is locked again. We will assume that no two residents have the same name.

To implement this program you will need the following classes:

1. **Door** class that simulates unlocking of the inner door
2. **Dorm** class that manages the resident information:
   * reading and saving the resident information from a file,
   * displaying the names of all the residents in the dorm,
   * retrieving information when given the resident's name,
   * it can also verify the validity of the entered data by checking them against the information kept by the Dorm’s registry.
3. **Resident** class that defines the Resident object, it throws IllegalArgumentException if the password is less than 3 characters long
4. **DormEntryClient** class has the main that interacts with the user

Implement all the classes and methods as defined in the UML diagram below. Please note that the skeleton classes are provided. Pay attention to the javadoc comments. The client is completed; however feel free to expand if necessary. The input file (residents.txt) containing the residents’ data is also provided.

Implement your program incrementally. Make sure to test your methods thoroughly. Please note that the sample runs of the finished application are provided below.

You must use Scanner for input, System.out for output, try-catch construct in the Dorm constructor to handle the IllegalArgumentException possibly thrown during instantiation of a Resident object.

**IMPORTANT:**

* Your code **must compile** in order to be graded.
* Each file **must be formatted** and **must contain your name.**
* Have only **one** return statement **per value returning method**
* Capture the **output** of your program and submit with the java files.
* When finished **zip** all the files and submit the package for grading**.**
* **Please do not remove //TODO comments from the code, it will help me with grading**

**UML diagram**:



**Sample run** of the program is given below:

\*\*\*Reading Data From The File\*\*\*

Missing data in record: Scott Adams:101; record ignored

Resident object created ---> Name: Alfred Aho, room: 102, password: AWK

Resident object created ---> Name: Paul Allen, room: 103, password: BASIC

Resident object created ---> Name: Tim Berners-Lee, room: 104, password: WWW

Resident object created ---> Name: Grady Booch, room: 105, password: UML

Resident object created ---> Name: Stephen Bourne, room: 201, password: BourneShell

Resident object created ---> Name: Vinton Cerf, room: 202, password: TCP/IP

Resident object created ---> Name: Ward Cunningham, room: 203, password: Wiki

Resident object created ---> Name: Edsger Dijkstra, room: 204, password: DijkstraAlgorithm

Resident object created ---> Name: Brendan Eich, room: 205, password: JavaScript

Resident object created ---> Name: Marc Ewing, room: 301, password: RedHatLinux

Resident object created ---> Name: Adele Goldberg, room: 302, password: Smalltalk

Resident object created ---> Name: James Gosling, room: 303, password: Java

Resident object created ---> Name: David Albert Huffman, room: 304, password: HuffmanCode

Missing data in record: Rebecca Heineman:305; record ignored

Resident object created ---> Name: Grace Hopper, room: 401, password: COBOL

Resident object created ---> Name: Brian Kernighan, room: 402, password: AWK

Resident object created ---> Name: Leslie Lamport, room: 403, password: LaTeX

Resident object created ---> Name: Rasmus Lerdof, room: 404, password: PHP

Resident object created ---> Name: Ada Lovelace, room: 405, password: firstProgrammer

Resident object created ---> Name: Yukihiro Matsumoto, room: 501, password: Ruby

Resident object created ---> Name: Scott Meyers, room: 502, password: C++

Resident object created ---> Name: Peter Naur, room: 503, password: ALGOL60

Resident object created ---> Name: Kristen Nygaard, room: 504, password: SIMULA

The password must be at least 3 characters long.: Dennis Ritchie:505:C; record ignored

Resident object created ---> Name: Guido van Rossum, room: 601, password: Python

Resident object created ---> Name: Cliff Shaw, room: 602, password: IPL

Resident object created ---> Name: Joel Spolsky, room: 603, password: StackOverflow

Resident object created ---> Name: Gerald Jay Sussman, room: 604, password: Scheme

Resident object created ---> Name: Niklaus Wirth, room: 605, password: Pascal

Resident object created ---> Name: Mark Zuckerberg, room: 701, password: Facebook

Resident object created ---> Name: Evan Spiegel, room: 702, password: Snapchat

Resident object created ---> Name: Kevin Systrom, room: 703, password: Instagram

Resident object created ---> Name: Jan Koum, room: 704, password: WhatsApp

Resident object created ---> Name: Larry Page, room: 705, password: Google

Missing data in record: Shel Silverstein:706; record ignored

\*\*\*End of input data\*\*\*

We have 32 residents in the dorm:

Alfred Aho; Paul Allen; Tim Berners-Lee; Grady Booch; Stephen Bourne; Vinton Cerf; Ward Cunningham; Edsger Dijkstra; Brendan Eich; Marc Ewing; Adele Goldberg; James Gosling; David Albert Huffman; Grace Hopper; Brian Kernighan; Leslie Lamport; Rasmus Lerdof; Ada Lovelace; Yukihiro Matsumoto; Scott Meyers; Peter Naur; Kristen Nygaard; Guido van Rossum; Cliff Shaw; Joel Spolsky; Gerald Jay Sussman; Niklaus Wirth; Mark Zuckerberg; Evan Spiegel; Kevin Systrom; Jan Koum; Larry Page;

\*\*\* Processing entry requests \*\*\*

Enter your name:

Anna Bieszczad

Anna Bieszczad is not a registered resident of this dorm

Is there another resident to enter the dorm? (yes/no)

yes

Enter your name:

Grace Hopper

Hello Grace Hopper, please enter your room number:

401

Enter password:

COBOL

The door is UNLOCKED

Is there another resident to enter the dorm? (yes/no)

yes

Enter your name:

Guido van Rossum

Hello Guido van Rossum, please enter your room number:

601

Enter password:

Python

The door is UNLOCKED

Is there another resident to enter the dorm? (yes/no)

yes

Enter your name:

James Gosling

Hello James Gosling, please enter your room number:

3003

Enter password:

Java

Invalid credentials

The door is LOCKED

Is there another resident to enter the dorm? (yes/no)

yes

Enter your name:

James Gosling

Hello James Gosling, please enter your room number:

303

Enter password:

java

Invalid credentials

The door is LOCKED

Is there another resident to enter the dorm? (yes/no)

yes

Enter your name:

James Gosling

Hello James Gosling, please enter your room number:

303

Enter password:

Java

The door is UNLOCKED

Is there another resident to enter the dorm? (yes/no)

no

Process finished with exit code 0