**California State Polytechnic University, Pomona**
**Computer Science Department**

[redacted]

Write an interactive Java program titled "KnightsOnABoard" which asks the user for the name of an input file. The input file consists of eight lines with 8 integer values of either 1 or 0 for each line as follows:

```
0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 0
0 0 0 0 1 0 1 0
0 1 0 0 0 1 0 0
0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1
0 0 0 0 1 0 0 0
```

Your program should open up the input file, read in every line and create an 8x8 2D array which contains this information. Make sure to do input validation at each step of the way. Is the input file a valid file? Is it indeed 8 lines of 8 integers? Is each integer a 1 or a 0? If the file is invalid, or if there are not 8 lines of 8 integers print an appropriate error message and ask the user to input a valid input file again (your program should continue to ask them to provide a valid input file until one is given). If the file has integer values other than 1 and 0, print an appropriate message alerting the user that positive values greater than 1 were detected and will be converted to 1 and/or that negative values smaller than 0 were detected and will be converted to 0. Do **not** print out a message if there are no problems with the input file and its data.

Once your program has a valid 8x8 2D array read in from the input file, we will use this data to represent knights on a chessboard where 1's represent where a knight is placed and 0's represent an empty square. Knights can be present in any of the 64 squares. No other pieces exist but knights. Given this understanding, write a method that returns true if the knights are placed on a chessboard in such a way that no knight can capture another knight. In the game of chess knights move in an L fashion (two spaces vertically and then one space horizontally, or two spaces horizontally and one space vertically) as shown below:

```
0 x 0 x 0
x 0 0 0 x
0 0 1 0 0
x 0 0 0 x
0 x 0 x 0
```

Here 1 is our knight, 0 is an empty square, and x is where our knight can move to.
Implement the following methods in the program:

- public File validateFile(File inputFile) – this method will check to see if the given file exists, if the input file does not exist it will give an appropriate error message and ask the user for another name until a valid one is given. It will then return the valid file.
- public boolean validateData(File inputFile) – this method will validate the data from the input file, or print an error message if there is not 8 lines of 8 integers. It will then return true if valid or false if invalid.

- public int[][] populateBoard(File inputFile) – this method will read through the file that has been validated, then create and populate an 8x8 2D array with the information from the file. Remember to correct any lines of data which are integers but not 1's or 0's and print an appropriate message if this is needed. It will then return the created array.
- public boolean cannotCapture(int[][] chessBoard) – this method computes if the knights are placed on the chessboard so that no knight can capture another knight. It will then return true if no knight can capture any other knight, or false if even one knight can capture another.
- public void printBoard(int[][] chessBoard) – this method will display the 2D array to the screen.
- public static void main (String[] args) – this is our standard main method to kick start the program. It will create an instance of our class then it will prompt the user for a file name and loop through validateFile and validateData until we can read in and create a valid 2D chess board array with populateBoard. It will then use printBoard to display the array and call cannotCapture and print an appropriate message to the screen depending on the true or false response.

The following shows an example interaction of running the program:
*NOTE* the first line is simply what you would type into the command line / terminal to run your program.

```
java KnightsOnABoard

Please enter the name of a valid file: file.txt
The board looks as follows:
0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 0
0 0 0 0 1 0 1 0
0 1 0 0 0 1 0 0
0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1
0 0 0 0 1 0 0 0

No knights can capture any other knight.
```

```
java KnightsOnABoard

Please enter the name of a valid file: invalid.txt
File has invalid data.
Please enter the name of a valid file: something
File does not exist!
Please enter the name of a valid file: file.txt
The board looks as follows:
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
0 0 0 0 1 0 1 0
0 0 1 0 0 1 0 1
1 0 0 0 1 0 1 0
0 0 0 0 0 1 0 1
1 0 0 0 1 0 1 0
0 0 0 1 0 1 0 1

There is at least one knight which can capture another knight.
```

What to turn in:
- Soft copy of the program (submit your .java file to Gradescope)