# Exercise 3

1. (a) Using BlueJ, create a project called co333Sudoku3.

* Add to this project the classes Sudoku and ProblemGrid from //raptor/courses/co333/1415/Sudoku3, and the Problem class from your own co333Sudoku2 project.
* Within this project, create a class called MyProblemGrid. Replace the class header with:   
    
  public class MyProblemGrid extends ProblemGrid  
    
  This makes MyProblemGrid a subclass of ProblemGrid. So MyProblemGrid inherits all fields and methods of ProblemGrid.
* MyProblemGrid is to contain a single method enterNumber. Enter the method header and bounding brackets for enterNumber in MyProblemGrid.
* Within your Problem class:
  + Use the constructor to initialize the elements of startValues array to 0.
  + Introduce a mutator called setStartValues. This method sets a specified element of the startValues array to the number supplied. Accordingly, it has three int parameters: the first to indicate the element’s row position, the second it’s column position and the third the value the element should be set to.

(b) In MyProblemGrid class, use your lecture notes to enter the Java code for the three methods: validEntry, obtainEntry and cellEntry.

* validEntry – This is the Boolean function that was constructed in exercise 2. It takes two parameters: the first a string parameter that represents the data entry; the second an int parameter that indicates the number of the cell where the data is being entered.
* obtainEntry – This is a static method provided by the abstract class ProblemGrid. It takes one int parameter that indicates the number of the cell where the data is being entered. See Javadoc of ProblemGrid.
* cellEntry - This is a static method provided the abstract class ProblemGrid. It takes three parameters: the first an int parameter that indicates the number of the cell where the data is to be entered; the second a string parameter that represents the data entry and the third a string parameter that indicates the required format. This third parameter is set to “blue” to display a blue font on a pink background or “black” to display a black font on a white background. This formatting is effected within the abstract class ProblemGrid. See Javadoc of ProblemGrid.

(c) Enter the code for enterNumber into your MyProblemGrid class in your BlueJ project and correct any syntax errors.

1. Testing: It is necessary to check that the user interface (GUI) works as intended and this can only be done manually.

Run through the test cases identified in exercise 2 for the input validating functions manually. In your log book, draw up a table that identifies the test case, outlines the purpose of the test, the required input data and the expected result. As each test is carried out, record whether the result is as expected and hence the test is passed or alternatively the test fails.

If a test fails, investigate and record the cause. Correct the error and retest from the beginning.