

COMP1406 - Assignment #1

(Due: Monday, January 20th @ 11:30pm)

In this assignment, you will create some simple programs in JAVA to get used to the language constructs that you learned last term.



(1) Guess A Number

Write a program (saved in a file called **GuessANumberProgram.java**) which randomly chooses an integer from 1 to 100. The program should then tell the user: "I am thinking of a number from 1 to 100 ... guess what it is ?". The user should then enter an integer as a *guess*. If the *guess* is above the randomly-chosen number, then the program should tell the user: "lower!" and then wait for another guess. If the *guess* was below the randomly-chosen number, then the program should tell the user: "higher!" and then wait for another guess. It should repeat this until the user enters the correct number. Then it should print out "Congratulations. You guessed the number with X tries!" where X is the number of guesses that the user made. The program should then quit. You may assume that the user always enters an integer. Here is example output:

```
I am thinking of a number from 1 to 100 ... guess what it is ?
34
higher!
78
higher!
86
lower!
82
Congratulations.  You guessed the number with 4 tries!
```

(2) The SMART Puzzle

Write a program (saved in a file called **SmartPuzzleProgram.java**)

which displays the following 5x5 table of letters. The program should then ask the user to complete the puzzle such that each row and each column consist of the letters 'S', 'M', 'A', 'R' and 'T' in some order. No letter should be repeated in any row or any column. The program should prompt the user for a row, a column and a letter. It should then insert (or replace) the letter at the given row and column with the entered letter and display the table again. If an invalid letter is entered the program should display "Invalid letter. Use 'S', 'M', 'A', 'R' or 'T'.". A placed letter can be replaced at any time. The program should repeatedly display the updated table and ask for a new letter to be placed. The program should end when each row and each column contains a unique valid letter. It should then display: "Congratulations! You must be SMART." and then quit. You MUST store all letters in a 2-dimensional array as follows:

		columns				
		1	2	3	4	5
rows	1	S	M	A	R	T
	2		T	S	M	
	3			R		S
	4		S	M		
	5			T	S	

```
char[][] table = new char[5][5];
```

Also, you **MUST** create a separate procedure for displaying the table and call it from your main program. The table should be displayed as shown below:

	1	2	3	4	5
0	S	M	A	R	T
1		T	S	M	
2			R		S
3		S	M		
4			T	S	

To determine if the game has ended, you can simply add up the numeric values for each letter (i.e., use `(int)table[r][c]` to get the numeric value for the letter at row `r` column `c`) in each row and column. The total for each row and column must be 391 exactly for the table to be completed properly.

NOTE: Submit all **.java** and **.class** files needed to run. You **MUST NOT use packages** in your code, **nor projects**. Submit ALL of your files in **one folder** such that they can be opened and compiled individually in JCreator. Some IDEs may create packages and/or projects automatically. You **MUST** export the **.java** files and **remove the package declaration** at the top if it is there. Do NOT submit JCreator projects either. **JUST SUBMIT the JAVA and CLASS FILES.** Note that if your internet connection at home is down or does not work, we will not accept this as a reason for handing in an assignment late ... so make sure to submit the assignment WELL BEFORE it is due !

Please NOTE that you WILL lose marks on this assignment if any of your files are missing. You will also lose marks if your code is not written neatly with proper indentation. See examples in the notes for proper style.

Sample program output:

```

      1   2   3   4   5
    ---+---+---+---+---
1 | S | M | A | R | T |
    ---+---+---+---+---
2 |   | T | S | M |   |
    ---+---+---+---+---
3 |   |   | R |   | S |
    ---+---+---+---+---
4 |   | S | M |   |   |
    ---+---+---+---+---
5 |   |   | T | S |   |
    ---+---+---+---+---
Enter a row (1-5): 3
Enter a column (1-5): 2
Enter a letter (S,M,A,R or T): G
Invalid letter. Use 'S', 'M', 'A', 'R' or 'T'.

```

```

      1   2   3   4   5
    ---+---+---+---+---
1 | S | M | A | R | T |
    ---+---+---+---+---
2 |   | T | S | M |   |
    ---+---+---+---+---
3 |   |   | R |   | S |
    ---+---+---+---+---
4 |   | S | M |   |   |
    ---+---+---+---+---
5 |   |   | T | S |   |
    ---+---+---+---+---
Enter a row (1-5): 3
Enter a column (1-5): 1
Enter a letter (S,M,A,R or T): T

```

```

      1   2   3   4   5
    ---+---+---+---+---
1 | S | M | A | R | T |
    ---+---+---+---+---
2 |   | T | S | M |   |
    ---+---+---+---+---
3 | T |   | R |   | S |
    ---+---+---+---+---
4 |   | S | M |   |   |
    ---+---+---+---+---
5 |   |   | T | S |   |
    ---+---+---+---+---
Enter a row (1-5): 4
Enter a column (1-5): 4
Enter a letter (S,M,A,R or T): T

```

```

      1   2   3   4   5
    ---+---+---+---+---
1 | S | M | A | R | T |
    ---+---+---+---+---
2 |   | T | S | M |   |
    ---+---+---+---+---
3 | T |   | R |   | S |
    ---+---+---+---+---
4 |   | S | M | T |   |
    ---+---+---+---+---
5 |   |   | T | S |   |
    ---+---+---+---+---
Enter a row (1-5):

```