

Software Workshop I
Assignment 2 - Practice!!!

Marks available: 20

Set by: Jacqui Chetty

Submission instructions

Submit your work by uploading the board-game.py as a zip file (only if you want to practice this).

Instructions:

1. Create a new Python project called
yourSurnameStudentNumber, for example Smith123456
2. Create a file called board-game.py
3. Copy the contents of the template given to you into
this file and make use of the instructions below to
complete the code for the game.
4. Do not put any code that gets user input. These values
must be passed as parameters - see template.
5. You may add more functions, but you may not remove
existing ones.
6. Please familiarise yourself with the template before
starting, especially taking note of the calling
statements found towards the end of the template.

Now you are ready to start your assignment solution:

1. **Go to the function generate_square_board() and create a
2 x 2 board based on the following criteria. [4]**
 - a. The board is always a square board with dimensions
of 2 x 2.
 - b. Make use of the variable square_board given in the
function and complete the necessary code required
to generate a board as a list (you won't see the
board yet). Each row / column should contain a
zero - see question 2 for an example.
 - c. Return the square_board.

2. Go to the function `print_board(square_board)` and print the board out in the following format: [5]

```
| 0 | 0 |  
| 0 | 0 |
```

- a. The `square_board` is passed in as a parameter.
- b. Each vertical line represents a separator.

3. Go to the function `generate_numbers(square_board)` where: [5]

- a. The `square_board` is passed in as a parameter.
- b. Generate a random number (import random) from 1 to 20 to replace each zero on the board, where the board could look as follows:

```
| 2 | 8 |  
| 3 | 4 |
```

- c. Return the `square_board`.

4. Go to the function `calculate_win(square_board)` and determine if the `square_board` produces a win based on the following rule: [6]

- a. A win is if the sum of all 4 numbers totals 10, 20, 30, 40, 50, etc.
- b. Return the message -> "There is a win" or "No win".
- c. For example, this window shows there is a win as the sum of the numbers add up to 40.

```
| 10 | 7 |  
| 15 | 8 |  
  
There is a win
```