UNIVERSITY of NICOSIA

COMP-113 Programming Principles II

Spring 2022

Instructor: Dr Athena Stassopoulou

Project #1 : Memory Game

OUT: Thursday, March 10, 2022.
DUE: Thursday, April 7, 2022 (source file on moodle, hard-copy in class).

No assignments will be accepted after this deadline.

INSTRUCTIONS:

1) Submit:

a) The .cpp file by uploading it at the course site (.cpp file ONLY). PLEASE name the file with your name as follows: AndreasAndreou.cpp

b) A hard copy of your code with the corresponding output screen printed. (NO LOOSE SHEETS OF PAPER WILL BE ACCEPTED. Submit everything neatly in a folder).

2) Your program should work with functions which should be provided to perform the various tasks.

You may only use the C++ programming concepts covered so far in class (i.e. up to and including Topic 4 Data Structures).

The examiner will attach great importance to the quality of code. Marks will be deducted for poor programming style. The program should be commented and indented to aid clarity.

NOTE: This is an individual assignment and should represent your own ideas, work and effort. In cases of plagiarism, a grade zero will be assigned to all parties involved.

a) Write a C++ program that plays the *memory game*. A description of the game is as follows: You have a 3x3 board in which you hide 4 pairs of letters which are directly coded into the program (one board position is blank). Each time, the player is asked to pick a pair of board positions. The contents of the board positions are then revealed and if the same letter appears in these positions, the player has successfully matched one of the 4 pairs. The process continues until the player has matched all 4 pairs. The program should allow a player to play again (if they wish), at the end of each game. (You can familiarize yourselves with an online version of the game at: http://www.brainmetrix.com/memory-game/).

A sample run should give the following (user's response shown here in **bold**):

```
Welcome to the Matching Game. At each move, choose two positions
that you think match.

Board is:
*   *   *
*   *   *
*   *   *

Board positions are:

1   2   3
4   5   6
7   8   9


Move #1: Enter two positions to match: 1 5
Board is:

b   *   *
*   c   *
*   *   *
```

[Screen is cleared after 2 seconds]

```
Board is:
*   *   *
*   *   *
*   *   *

Move #2: Enter two positions to match: 1 3
Board is:

b   *   b
*   *   *
*   *   *

Move #3: Enter two positions to match: 4 5
```

```
 Board is:
b  *  b
   c  *
*  *  *

 [Screen is cleared after 2 seconds]


 Board is:
 b  *  b
 *  *  *
 *  *  *

 Move #4: Enter two positions to match: ….
 …..
 ….
 Board is:
 b  d  b
    c  d
 a  c  a

 You did it in 7 moves!  Play again (Y or N)? N
 Bye.
```

(b) Modify your program in (a) above so that the letters are allocated randomly within the 3x3 array. You will need a random number generator that randomly chooses the positions of the pairs. There are 4 pairs of letters (a, b, c, and d) and one board position will be left blank. You will call the random number generator at least twice for each pair to get the two positions. (You may need to call it more than twice if it gives you a position which is already taken). The random number generator should generate numbers from 1 to 9 (inclusive), indicating the board positions. The following C++ statement issues an integer form 1 to 9 and assigns it to an integer `r`: `r=rand()%9 +1;` (See Random Number Generators topic).

If you get to this point successfully, you only need to submit a printout of this final version.

Notes:

- You will need to use two-dimensional arrays.
- To clear the screen, simply print a sufficient number of end lines.
- To make the program pause some seconds before you clear the screen, you may use the function Sleep() which takes as a parameter an integer which is the number of milliseconds that it will wait. E.g. the statement: `Sleep(2000);` causes a time delay (i.e. a pause) of 2000 milliseconds. In order to use this function, you will need to include the header file <Windows.h>.