CPSC 21000: Programming Fundamentals
Programming Assignment 3 – MouseRace

Introduction

For this assignment, you are to create a program in Java that implements the Mouse Race game. The game will proceed as follows:

- You should first display header information (course name, semester, name, assignment#) and the message: "WELCOME TO THE MOUSE RACE GAME".

- Then enter the *game loop*. In the game loop, you will repeat the following until the game ends:

  - Display the maze. The maze is a 10x10 grid of characters, with a "wall" around it (so total size is 12x12). This maze should look exactly like the one in the sample output (see last section), except the location of the mouse might be different depending on which moves were made. Represent the mouse as '%' and the cheese as '$'. The mouse will always begin in the upper left corner and the cheese will always be in the lower right corner.

  - Check if the game ended. The game ends when the location of the mouse is the same as the location of the cheese. If the game ends, quit the loop.

  - Display the current score. The score is initially 0. Each move costs 1 point (subtract 1 from the score). Getting the cheese is +100 points.

  - Prompt for direction. The directions are 'n' for north, 's' for south, 'w' for west, and 'e' for east. If the move is chosen and there is no wall there, change the location of the mouse to the corresponding location. Otherwise (if the mouse bumps into a wall), the location of the mouse will not change and you should display the message "You cannot move there!".

- Once the game ends, display the message "GAME OVER! MOUSE GOT THE CHEESE!" and "Your score was ...", replacing the ... with the actual score after the game ended.

Program Design Requirements

- You should have a single class called `MouseRace`.
- Inside this class you will have static variables that correspond to things you need to keep track of in the game: the location of the mouse (row and column), the location of the cheese (row and column), the maze itself (2-D array of `char`s), and the score.
- You should have at least 6 methods:

1. `main`: should only display the header information and call the startGame() method.
2. `startGame`: implements the game - initializes the variables, displays the intro message, and then enters the game loop. It will call other methods to print the board, check for a game end, etc.
3. `gameWon`: should return true if the game was won and false otherwise.
4. `printMaze`: should print the maze (including the mouse and cheese).
5. `makeMove`: should attempt to make the move to the specified row, column.
6. `parseCmd`: should accept a string and call the `makeMove` method with the appropriate parameter values. For example, if the command is "n", then it should call `makeMove(mouseLocRow-1, mouseLocCol)`.

Additional Requirements

1. The name of your Java Class that contains the main method should be `MouseRace`. All your code should be within a single file.
2. Your code should follow good coding practices, including good use of whitespace (indents and line breaks) and use of both inline and block comments.
3. You need to use meaningful identifier names that conform to standard Java naming conventions.
4. At the top of the file, you need to put in a block comment with the following information: your name, date, course name, semester, and assignment name.
5. The output of your program should **exactly** match the sample program output given at the end.

What to Turn In

You will turn in the single MouseRace.java file using BlackBoard.

What You Need to Know for This Assignment

- Conditional statements
- Loops
- Getting user input
- Creating and calling methods
- Difference between static class variables and local variables
- Arrays

Sample Program Output

```
CPSC 21000
NAME: <name>
PROGRAMMING ASSIGNMENT 3

**************************************
*** WELCOME TO THE MOUSE RACE GAME ***
**************************************

############
#% ####### #
## ######  #
##     ## ##
#  ### #  ##
# #### #  ##
# #  # ## ##
# ## # ## ##
#  # # ## ##
## # #    ##
#    #### $#
############

Your current score: 0
Select a move direction (n/s/w/e):n

You cannot move there!


############
#% ####### #
## ######  #
##     ## ##
#  ### #  ##
# #### #  ##
# #  # ## ##
# ## # ## ##
#  # # ## ##
## # #    ##
#    #### $#
############

Your current score: -1
Select a move direction (n/s/w/e):e
```

```
############
# %####### #
## ######  #
##     ## ##
#  ### #  ##
# #### #  ##
# #  # ## ##
# ## # ## ##
#  # # ## ##
## # #    ##
#    #### $#
############

Your current score: -2
Select a move direction (n/s/w/e):s


############
#  ####### #
##%######  #
##     ## ##
#  ### #  ##
# #### #  ##
# #  # ## ##
# ## # ## ##
#  # # ## ##
## # #    ##
#    #### $#
############

Your current score: -3
Select a move direction (n/s/w/e):
```

…
…
…
SAMPLE PROGRAM OUTPUT cont. (shows just the game ending)

```
############
#  ####### #
## ######  #
##     ## ##
#  ### #  ##
# #### #  ##
# #  # ## ##
# ## # ## ##
#  # # ## ##
## # #    ##
#    ####%$#
############

Your current score: -19
Select a move direction (n/s/w/e):e
```

```
############
#  ####### #
## ######  #
##     ## ##
#  ### #  ##
# #### #  ##
# #  # ## ##
# ## # ## ##
#  # # ## ##
## # #    ##
#    #### %#
############
GAME OVER! MOUSE GOT THE CHEESE!
Your score was 80
```