Make sure your prompts and printouts match EXACTLY to the specification (INCLUDING capitalization, punctuation, spaces and newlines)!

Similarly, your java files / class names / method names MUST match the spelling and capitalization EXACTLY!

With the exception of constants, NO DATA IN YOUR CLASS SHOULD BE PUBLIC OR DEFAULT/PACKAGE VISIBLE. Use private or protected!

You can use the checkboxes to track whether you've met each requirement.

#	*Requirements*	
1	**Create a class named MapCoord with the following public NON-STATIC instance methods:**	
	public MapCoord(int r, int c)	
	This CONSTRUCTOR takes the row and column for the particular coordinate and stores it in the class.	
	You will mostly likely want to have instance variables for the row and column as integers.	
	public int getRow()	
	Returns the stored row value.	
	public int getColumn()	
	Returns the stored column value.	
	public String toString()	
	Returns a String with "(" + the stored row coordinate + "," + the stored column coordinate + ")" For example, row = 4 and column = 12 should be: "(4,12)"	
	This does NOT print anything! This ONLY returns a String!	

2	**Create a class named TextMap with the following public NON-STATIC instance methods:**	
	public TextMap(int rowCnt, int colCnt, char defaultChar)	
	This CONSTRUCTOR should create a 2D array of characters with *rowCnt* rows and *colCnt* columns, filled with the *defaultChar* passed in.	
	You will mostly likely want to store the row count, column count, default character, and the 2D character array as instance variables.	
	public int getRowCnt()	
	Returns the total number of rows.	
	public int getColumnCnt()	
	Returns the total number of columns.	
	public boolean isValidPosition(MapCoord coord)	
	Returns true if the coordinate passed in is a valid location on the map (e.g., within bounds).	
	Returns false if the coordinate is out of bounds.	
	public char getPos(MapCoord coord)	
	If the coordinate is a valid position, returns the character at that location.	
	Otherwise, returns the default character.	
	public boolean setPos(MapCoord coord, char c)	
	If the coordinate is a valid position, stores character *c* in the specified coordinate position in the array, and then returns true.	
	Otherwise, does nothing to the map and returns false.	

	public String toString()	
	Returns a String representing the map data.	
	Note that EVERY row of the map should end in a newline character (including the last row).	
	See below for an example of what a printout of these Strings look like, BUT do NOT put SPACES between characters!	
	This does NOT print anything! This ONLY returns a String!	
3	**Create a class named MapEditor with a main() method that does the following:**	

	Create a Scanner object to read from System.in.	
	Print out "Enter rows, columns, and default character:" using System.out.println().	
	Using your Scanner object, read in the row count and column count as integers using nextInt(), and the default character (using next().charAt(0) on your Scanner object to read in a single character).	
	Create an instance of TextMap with the specified row count, column count, and default character.	
	Print the map.	
	If toString() is defined correctly, you can actually do the following: `System.out.println(myMap);` ...where myMap is an instance of TextMap.	
	Do the following while the user enters a valid position:	
	Print out "Enter row, column, and character:" using System.out.println().	
	Using your Scanner object, read in the row and column as integers using nextInt(), and a character (using next().charAt(0) on your Scanner object to read a single character).	
	Call setPos() on your TextMap object, passing in the specified row, column, and character. You will have to create an instance of MapCoord to pass in the row and column.	
	Print the current map (regardless of whether the user entered a valid coordinate or not).	

Sample Run for MapEditor

```
Enter rows, columns, and default character:
5 10 .
..........
..........
..........
..........
..........

Enter row, column, and character:



 0 4 #
 ....#.....
 ..........
 ..........
 ..........
 ..........

 Enter row, column, and character:
 3 1 @
 ....#.....
 ..........
 ..........
 .@........
 ..........

 Enter row, column, and character:
 9 0 $
 ....#.....
 ..........
 ..........
 .@........
 ..........
```

NOTE: ONLY MapEditor has a main() method!