CECS 277 – Lab 5 – Classes & Objects

Moving Rectangle

Create a program that allows the user to move a rectangle around a grid. The user should input the dimensions of the rectangle (width and height 1-5), and then be able to choose which direction to move the rectangle in.

Create a Rectangle class (in a separate file named 'rectangle.py') with the following:

1. Attributes – integers `x`, `y`, `width`, `height`
2. `__init__(self, w, h)` – pass in w and h (not x and y), assign them to width and height, set the x and y attributes to 0.
3. `get_coords(self)` – returns the x and y values as a pair (either a tuple or a list).
4. `get_dimensions(self)` – returns the rectangle's width and height as a pair.
5. `move_up(self)` – moves the rectangle up one row.
6. `move_down(self)` – moves the rectangle down one row.
7. `move_left(self)` – moves the rectangle left one column.
8. `move_right(self)` – moves the rectangle right one column.

In a separate file ('main.py'), in the main function, prompt the user to enter a width and height of a rectangle (1-5). Use these inputs to create an instance of the rectangle. Create a 20x20 2D list (the grid) that is initially all '.'s.

Also create the following functions in your main.py:

1. `display_grid(grid)` – pass in the grid and display the contents of the grid.
2. `reset_grid(grid)` – pass in the grid and overwrite the contents with all '.'s.
3. `place_rect(grid, rect)` – pass in the grid and the rectangle. At the location of the rectangle (x, y) on the grid, overwrite the '.' characters with '*'s using the width and height of the rectangle.

After the rectangle is constructed, place it on the grid, display the grid, and then display a menu that allows the user to choose which direction to move the rectangle in, up, down, left, right, or to quit. Once the user has chosen a direction, check that the rectangle can move in that direction, it should not be able to move past the boundaries of the grid in any direction and should display an error message. If the rectangle has room to move, then you should call the move method for that direction to update the rectangle's location. Then call the reset_grid, place_rect, and display_grid functions to update and display the grid. Repeat until the user chooses to quit the program. Use the check_input module to validate all user inputs. Use docstrings to document the class, its methods, and the main functions.

Example Output (user input is in italics):

```
Enter rectangle width (1-5): 4
Enter rectangle height (1-5): g
```

```
Invalid input – should be an integer.
Enter rectangle height (1-5): 7
Invalid input – should be within range 1-5.
Enter rectangle height (1-5): 3

* * * * . . . . . . . . . . . . . . . . .
* * * * . . . . . . . . . . . . . . . . .
* * * * . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .

Enter Direction:
1. Up
2. Down
3. Left
4. Right
5. Quit
2

. . . . . . . . . . . . . . . . . . . . .
* * * * . . . . . . . . . . . . . . . . .
* * * * . . . . . . . . . . . . . . . . .
* * * * . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . .
```

```
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Enter Direction:
1. Up
2. Down
3. Left
4. Right
5. Quit
5
```

Notes:

1. You should have 3 different files in your project: rectangle.py, check_input.py, and main.py.
2. Check all user input using the get_int_range function in the check_input module.
3. Do not create any extra attributes, methods, parameters, or functions.
4. Please do not create any global variables or use the attributes globally. Only access the attributes using the class's methods.
5. Use docstrings to document the class, each of its methods, and the functions in main.py. See the lecture notes and the Coding Standards reference document for examples.
6. Place your names, date, and a brief description of the program in a comment block at the top of your main. Place brief comments throughout your code.
7. Thoroughly test your program before submitting:
    a. Make sure that all user inputs are validated.
    b. Make sure that the rectangle is drawn with the correct dimensions (ie. make sure that you didn't get the width and height backwards), and in the correct location.
    c. Make sure that the rectangle moves in the direction that the user selects.
    d. Make sure that the rectangle cannot move past the border of the grid. Up and left should check the top and left edges of the rectangle, and down and right should check the bottom and right edges of the rectangle (which means you'll also have to account for the width and height of the rectangle).

Moving Rectangle Rubric – Time estimate: 3 hours

Moving Rectangle **10 points**	Correct. 2 points	A minor mistake. 1.5 points	A few mistakes. 1 point	Several mistakes. 0.5 points	No attempt. 0 points
Rectangle class: 1. Created in a separate file. 2. Has attributes x, y, width, height. 3. Has methods: __init__, get_coords, get_dimensions, and four moves. 4. Move methods correctly update the x, y coordinates. 5. Attributes are only accessed through the methods and not directly.					
reset_grid and display_grid functions: 1. Pass in grid, does not return. 2. reset_grid overwrites contents of grid with all '.'s. 3. display_grid prints contents as a 20x20 grid.					
place_rect function: 1. Passes in grid and rectangle. 2. Places '*'s in the grid at the location of the rectangle as a box of width x height. 3. Does not return the grid.					
Main Function: 1. Prompts user for rect size. 2. Constructs rectangle object. 3. Creates 20x20 grid. 4. Prompts user to move rect. 5. Moves rectangle in the proper direction. 6. Rectangle does not move out of bounds of grid (displays error msg). 7. Updates and displays grid. 8. Validates all user input. 9. Repeats until user quits.					
Code Formatting: 1. Code is in functions. 2. Correct spacing. 3. Meaningful variable names. 4. No global variables or accessing attributes directly. 5. Correct documentation.					