# CS 5000 – Summer 2025
# Assignment #9, 50 Points

## Classes and Objects – Chapter 10

**Remember to read and apply the good coding practices outlined in the previous assignment.**

Develop a complete Java program for each of the following problems. Please name the programs as indicated and add proper program headers and output labels as shown below. ***Please use only concepts and programming constructs/syntax we discuss to date.***

**Program #1 (50 points):** Design and implement a Java class, named ***HotelRating***, to implement the concept of rating hotels over a number of years. The rating is represented by a number of stars: one star (lowest quality) to five stars (highest quality). The only data field the class has is a two-dimensional array of values representing the quality ratings. Dimension 1 (rows) represents the hotels and dimension 2 (columns) represents the years. The class constructor method takes two parameters representing the number of hotels and the number of years. These values decide the size of the array (rows x columns). The ratings are randomly generated integer values (between 1 and 5 inclusive) and stored in the array. The class defines the following methods:

1. A method named `fiveStarsHotels()` to return an array of the indices (index values) of hotels that have earned five stars at least once over the years.
2. A method named `averageRatings()` to printout the average rating (double) for each hotel over the years.
3. A method `fourOrMoreStars()` to printout the indices (index values) of hotels that have earned four or more stars at least once.
4. A method named `anyFiveStars()` to return *true* if at least one hotel earned five stars for at least one year; *false* otherwise.
5. A method named `bestHotel()` to return the index for the best quality hotel over the years. If more than one hotel, return the index of the first hotel in the array.
6. A method named `worstHotel()` to return the index for the worst quality hotel over the years. If more than one hotel, return the index of the first hotel in the array.
7. A method named `printChart()` to printout the ratings for all hotels as shown below (for illustration, assume we have 3 hotels and 4 years of ratings):

```
              Year1       Year2       Year3       Year4
-------------------------------------------------------------------------------
Hotel 0:    **          ***         *****       ***
Hotel 1:    ***         ***         **          **
Hotel 2:    *****       ****        ****        *****
```

Write a test program in a separate file, named **`TestHotelRating`**, to create an object of the class and test all seven class methods on that object following the sample run below. Allow the user to enter both number of years and number of hotels. Do not hard-code the array size. Use a sentinel loop to allow re-runs and re-create the object with different input values. Document your code, organize, and space the outputs properly. Use escape characteristics and formatting objects ($ and %) as needed. The following sample test shows only the output labels and format (assuming three hotels and four years for illustration purpose). Make sure your code displays the outputs following the test data format.

Test data:

```
Five stars hotels indices:            0, 2
Average Ratings are:
        Hotel 0:                      3.25
        Hotel 1:                      2.25
        Hotel 2:                      4.50
Four or more stars hotels indices:    0, 2
Any five stars hotel?                 true
Best hotel index:                     2
Worst hotel index:                    1
```

```
Ratings chart:

                Year1       Year2       Year3       Year4
-------------------------------------------------------------------------------
Hotel 0:        **          ***         *****       ***
Hotel 1:        ***         ***         **          **
Hotel 2:        *****       ****        ****        *****

Would you like to run the program again (y/n)? y
```

## Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines posted in D2L. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment due date is posted in D2L.
3. Please compile, run, and test your code right before you upload your java files to the assignment submission folder in D2L.
4. Please upload only the .java files (total 2 files).