CS2163 Java Homework 9 requirement

Homework 9 needs to be finished on the Greenfoot platform. Homework 9 does not involve Eclipse.

**Follow the steps below to finish homework 9:**

* work on these two exercises in textbook: 9.69 and 9.70. It requires you to open the existing Greenfoot scenario folder “asteroids-3”, and then modify its source code.
* provide the solution by modifying class Space and class Asteroid. No other class needs to be modified.
* for exercise 9.69, you need to read the source code in class Counter, and find out which method to call in class Counter, in order to add a new score to the current counter value. This should be very straightforward.
* for exercise 9.70, you may need to consider making use of the generic class type List<>. In the previous homework, hw #8, you have applied all OOP principles and all the context of polymorphism, except for generic programming (static binding). And in this homework, hw #9, you will have a chance to use generic programming, such as List<>. With homework #8 and #9, you have applied all the OOP principles and context into your won coding practice.
* your solution needs to be efficient. For example, in exercise 9.70, you should insert the block of code of calculating how many asteroids are left in the world background, only at the place where it is necessary. If your code of calculating how many asteroids are left is placed directly in the act() method of a class, that means in every act() cycle your code of calculating remaining asteroids will run, and that is not considered efficient.
* you should have reasonable scores for splitting asteroid and removing the last piece, respectively. For example, there should be a higher score for removing the last piece of an asteroid, comparing to splitting an asteroid into smaller pieces.
* to satisfy the requirement of exercise 9.69, you need to update the score counter on the lower left corner of the screen, and you need to display the correct score value in the final score screen. This final score screen will show up after the rocket collides with any asteroid thus causing the game to end.
* you can finish exercise 9.69 first, and then work on 9.70. The solution of them should co-exist in one project, and you don’t need to comment out the solution for one or the other exercise.
* after finish both exercises, zip the entire scenario folder “asteroids-3”, and rename the zip file as “***JohnDoeHw9.zip***”, where JohnDoe needs to be replaced by your first and last name
* submit the zip file ***JohnDoeHw9.zip*** to Moodle “homework 9 drop box”.

**After finishing this homework, how to verify the correctness of your submitted zip files:**

1. Download the zip files you have uploaded to Moodle homework drop box.
2. Unzip the zip file to a different local folder in your computer, other than the original local folder where the zip files are generated.
3. Run the Greenfoot project from the unzip folder, and make sure it compiles and runs correctly.
4. If your submitted zip file in the Moodle drop box

**cannot be downloaded,** or

**cannot be unzipped,** or

**cannot compile,** or

**cannot run,**

then you need to figure out the reason and fix the error, and then submit the corrected zip file to the Moodle drop box. Then start this verification process again until you can download, unzip, compile and run successfully. To upload a corrected zip file to the Moodle drop box, you need to delete the previous submitted zip file from the Moodle drop box first.

In the first page of file “chap1-schedule.docx”, you can find the instructions on how to zip and unzip files.

**Grading components:**

* Each exercise has 10 points, thus results in a total of 20 points for this homework

**For any submitted zip file that still has syntax error and it cannot compile or run in Greenfoot, it will receive ZERO point**. No re-submission is allowed after the homework due day.

Please click the Moodle homework drop box to see the due day of this homework.

When coding in Eclipse and Greenfoot, please read document “RulesForIndentAndAlignCode.docx” in Moodle folder “chap 1”, and follow all the rules in code alignment and indentation.