

Assignment 1: Superhero Tracker (15% of course total)

- Due date is **5 June, 11:59pm**. Refer to Canvas for any updates.
- Submit deliverables to the corresponding assignment folder on Canvas.
- Make sure you also complete the corresponding online quiz at CourSys to receive full portion of the coding part. Refer to Course Outline and Marking Guide for details,
- Late penalty is 10% per calendar day (each 0 to 24 hour period past due), max. 2 days late (20%)
- This assignment is to be done individually. Do not show another student your code, do not copy code from another person/online. Please post all questions on Canvas.
- You may use general ideas you find online and from others, but your solution must be your own.
- You may use any tool for development. But we will be grading your submission using IntelliJ IDEA Community, so make sure your code compiles and runs there. We support only IntelliJ.

Description

In this assignment you are going to implement a text-based query system of superheroes. This system will allow a user to maintain a list of superheroes recording a small collection of their attributes.

Overall Requirements

You must have (at least) the following three classes (in separate .java files):

- A class for holding superhero information: name (String), superpower (String), height (double, in centimetres, not inches), and number of civilians saved (int)
 - Name may contain more than one word (e.g., "Green Lantern"), so is superpower (e.g., "can fly").
 - This class must correctly implement toString(), as discussed in lecture.
- A class for a text menu
 - Have a field to store the menu's title (String).
 - Have a field to store the menu's options (as an array of Strings).
 - The menu option strings should not include numbers. So instead of having it store {"1. Do thing one", "2. Do thing two"}; have it store {"Do thing one", "Do thing two"} and your program generate the numbers automatically.
 - Have a method to display (print) the menu to the screen.
 - Your program must automatically place a rectangle of *'s around the menu's title, sized to the length of the title. This must be computed, not hard-coded!
 - Automatically number the options starting at 1.
- A class for the main application
 - Contains a main() method which uses the menu and superhero classes to implement the application. It should read the user input and call upon appropriate methods from them.
 - Creates an ArrayList of superheroes to hold the list of superheroes the user enters.
 - Be careful not to have much duplicate code in your application! Use methods.

For this assignment, it is fine if all your classes are in one package.

Your system must be able to save the superhero-list in JSON format (e.g., a JSON array object). Upon the first time running the system, this list is empty (because no superheroes have been added). When the user

exits the system, a JSON file is automatically generated and saved in your project folder (using a hard-coded relative path). When the user runs the system the second time (and subsequent times), this JSON file will be loaded and used to populate the superhero-list. When the user exits again, this file will be updated with the new information. Both the loading and saving are done automatically by the program – the user doesn't need to do anything to trigger that.

Text Interface Requirements

- When you prompt the user to choose a menu option, if the user enters an invalid number you must re-ask the user to enter a valid value.
 - You may assume the user always enters correct type of data: when asked for an int, it is OK if the program crashes when the user enters a non-int such as 'A'.
- **Main Menu Option 1: List all superheroes**
 - For each superhero, show in a single line including: hero number, name, superpower, height, and number of civilians saved.
 - The hero number is simply a counter of the listed superhero (starting with 1).
- **Main Menu Option 2: Add a new superhero**
 - Ask user for name, superpower, and height of the superhero. Set the number of civilians saved to zero to start. Handle the case when a non-positive number is inputted as height.
 - The new superhero is always added to the end of the list.
 - When done, display an acknowledge message. For example, if “Batman” is added, display “Batman has been added to the list of superheroes.”.
- **Main Menu Option 3: Remove a superhero**
 - List all superheroes currently in the system. Ordered by when they are added to the list.
 - Allow the user to select a superhero (by number), or 0 to cancel.
 - Entering an invalid number (like -3, or a value more than the number of superheroes) needs to be handled by the application (tell the user that it is invalid and ask for a number again). Entering invalid data type (“hello”) does not need to be handled.
 - When done, display an acknowledge message. For example, if “Superman” is removed, display “Superman has been removed from the list of superheroes.”.
- **Main Menu Option 4: Update number of civilians saved by a superhero**
 - Similar to “remove a superhero”, user selects a superhero to work with (or 0 to cancel).
 - Update the number of civilians save by the selected superhero.
 - Entering an invalid number (like -3) needs to be handled by application (tell the user that it is invalid and as for a number again). Entering invalid data type (“hello”, “2.5”) does not need to be handled.
 - When done, display an acknowledge message. For example, if “Captain Marvel” has saved 1000 more civilians from the original 2000, display “Number of civilians saved by Captain Marvel has been updated from 2000 to 3000.”.
- **Main Menu Option 5: List top 3 superheroes**
 - List the top 3 superheroes who have saved the most civilians. For each superhero, show the name and then the number of civilians saved in a separate line. For example:
 - Superman saved 345000 civilians
 - Batman saved 234000 civilians
 - Wonder Woman saved 123000 civilians

- To be on top 3 list the superhero must have saved at least one civilian. If there is less than 3 to show, display this message instead “The superheroes have not saved enough civilians.”. If there is a tie, any order is acceptable. If the 3rd and 4th are the same, anyone is ok.
- In case there is less than 3 entries in the superhero-list, display this message instead “There is not enough superheroes in the list. Please add more.”.
- **Main Menu Option 6: Debug dump of superhero details**
 - For each superhero in the superhero-list, call toString() on each superhero and print the result to the screen.
- **Main Menu Option 7: Exit**
 - Display the message “Thank you for using the system.”, then exit the application.
 - It will also automatically save the superhero-list in JSON format.

Your text UI need not match the sample exactly, but it should be of equal quality.

Coding Requirements

- Your code must conform to the programming style guide for this course; see course website.
- All classes must have a class-level JavaDoc comment describing the purpose of the class.

Suggestions

- Think about the design before you start coding
 - List the classes you expect to create.
 - For each class, decide what its responsibilities will be.
 - Think through some of the required features. How will each of your classes work to implement this feature? Can you think of design alternatives?
- Write your code in progressing level of details
 - Your code does not need to be fully functional at the first time. Start with just printing a message in a method to have the logic correct.
 - Use refactoring to improve your code in later stages.
- You can reuse some of the methods to save time. For example, think about how to reuse the “List all superhero” option in the “Remove superhero” option.

Submission Instructions

- Submit a ZIP file of your project to the corresponding folder on Canvas. Name it using this format: **<firstname_lastname>_<studentID>_Assignment1.zip**. See course website for directions on creating and testing your ZIP file for submission. If you have any difficulties in submitting your file on Canvas, email it to the instructor.
- Save a copy in a secure location for safe keeping and do not modify it after the deadline.
- If you use any libraries they have to be included in the project as well.
- All submissions will automatically be compared for unexplainable similarities.

Useful Resources

- The GSON library for handling JSON: <https://github.com/google/gson>
- A video tutorial for File I/O with GSON by Dr. Brian Fraser: <https://youtu.be/HSuVtkdej8Q>
- Height Scale for Marvel Characters: https://marvel.fandom.com/wiki/Height_Scale
- Height Scale for DC Characters: <https://dc.fandom.com/wiki/Category:Height>

Sample Input/Output

(first time running the system)

```
*****
SuperHero Tracker
*****
1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 1
No heroes present in the list. Add more..

*****
SuperHero Tracker
*****
1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 2
Enter Hero's name: Captain Marvel
Enter Hero's height in cm: 170
Enter Hero's Superpower: Photo Blast
Captain Marvel has been added to the list of superheroes.

*****
SuperHero Tracker
*****
1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 2
Enter Hero's name: Batman
Enter Hero's height in cm: 183
Enter Hero's Superpower: Super Rich
Batman has been added to the list of superheroes.
```

SuperHero Tracker

1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 2
Enter Hero's name: Earthworm Jim
Enter Hero's height in cm: 173
Enter Hero's Superpower: Superior Strength
Earthworm Jim has been added to the list of superheroes.

SuperHero Tracker

1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 2
Enter Hero's name: Storm
Enter Hero's height in cm: 200
Enter Hero's Superpower: Control Weather
Storm has been added to the list of superheroes.

SuperHero Tracker

1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 1
1. Hero name: Captain Marvel, height: 170.0cm, superpower: Photo Blast, saved 0 civilians.
2. Hero name: Batman, height: 183.0cm, superpower: Super Rich, saved 0 civilians.
3. Hero name: Earthworm Jim, height: 173.0cm, superpower: Superior Strength, saved 0 civilians.
4. Hero name: Storm, height: 200.0cm, superpower: Control Weather, saved 0 civilians.

SuperHero Tracker

- 1: List all superheroes
- 2: Add a new superhero
- 3: Remove a superhero
- 4: Update number of civilians saved by a superhero
- 5: List Top 3 superheroes
- 6: Debug Dump (toString)
- 7: Exit

Enter >> 3

1. Hero name: Captain Marvel, height: 170.0cm, superpower: Photo Blast, saved 0 civilians.
2. Hero name: Batman, height: 183.0cm, superpower: Super Rich, saved 0 civilians.
3. Hero name: Earthworm Jim, height: 173.0cm, superpower: Superior Strength, saved 0 civilians.
4. Hero name: Storm, height: 200.0cm, superpower: Control Weather, saved 0 civilians.

Enter Hero number to be removed or Enter 0 to cancel

Enter >> 3

Earthworm Jim has been removed from the list of superheroes.

SuperHero Tracker

- 1: List all superheroes
- 2: Add a new superhero
- 3: Remove a superhero
- 4: Update number of civilians saved by a superhero
- 5: List Top 3 superheroes
- 6: Debug Dump (toString)
- 7: Exit

Enter >> 5

The superheroes have not saved enough civilians.

SuperHero Tracker

- 1: List all superheroes
- 2: Add a new superhero
- 3: Remove a superhero
- 4: Update number of civilians saved by a superhero
- 5: List Top 3 superheroes
- 6: Debug Dump (toString)
- 7: Exit

Enter >> 4

1. Hero name: Captain Marvel, height: 170.0cm, superpower: Photo Blast, saved 0 civilians.
2. Hero name: Batman, height: 183.0cm, superpower: Super Rich, saved 0 civilians.
3. Hero name: Storm, height: 200.0cm, superpower: Control Weather, saved 0 civilians.

```
Enter Hero number to update civilian count or Enter 0 to cancel
Enter >> 1
Enter new civilian save count: 5000
Number of civilians saved by Captain Marvel has been updated from 0 to 5000

*****

SuperHero Tracker
*****

1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 4
1. Hero name: Captain Marvel, height: 170.0cm, superpower: Photo Blast, saved 5000 civilians.
2. Hero name: Batman, height: 183.0cm, superpower: Super Rich, saved 0 civilians.
3. Hero name: Storm, height: 200.0cm, superpower: Control Weather, saved 0 civilians.

Enter Hero number to update civilian count or Enter 0 to cancel
Enter >> 2
Enter new civilian save count: 300
Number of civilians saved by Batman has been updated from 0 to 300

*****

SuperHero Tracker
*****

1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 4
1. Hero name: Captain Marvel, height: 170.0cm, superpower: Photo Blast, saved 5000 civilians.
2. Hero name: Batman, height: 183.0cm, superpower: Super Rich, saved 300 civilians.
3. Hero name: Storm, height: 200.0cm, superpower: Control Weather, saved 0 civilians.

Enter Hero number to update civilian count or Enter 0 to cancel
Enter >> 3
Enter new civilian save count: 2000
Number of civilians saved by Storm has been updated from 0 to 2000

*****

SuperHero Tracker
*****
```

```
1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 5
1. Captain Marvel has saved 5000 civilians
2. Storm has saved 2000 civilians
3. Batman has saved 300 civilians

*****

SuperHero Tracker
*****

1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 6
1. MarvelHero{name='Captain Marvel', heightInCm=170.0, civilianSaveCount=5000,
superPower='Photo Blast'}
2. MarvelHero{name='Batman', heightInCm=183.0, civilianSaveCount=300, superPower='Super Rich'}
3. MarvelHero{name='Storm', heightInCm=200.0, civilianSaveCount=2000, superPower='Control
Weather'}

*****

SuperHero Tracker
*****

1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 7
Thank you for using the system.
```

(Second time running the system, note how the superhero-list is populated with the superheroes adding the in the previous run)

```
*****

SuperHero Tracker
```



```
*****
1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 1
1. Hero name: Captain Marvel, height: 170.0cm, superpower: Photo Blast, saved 5000 civilians.
2. Hero name: Batman, height: 183.0cm, superpower: Super Rich, saved 300 civilians.
3. Hero name: Storm, height: 200.0cm, superpower: Control Weather, saved 2000 civilians.

*****
SuperHero Tracker
*****
1: List all superheroes
2: Add a new superhero
3: Remove a superhero
4: Update number of civilians saved by a superhero
5: List Top 3 superheroes
6: Debug Dump (toString)
7: Exit
Enter >> 7
Thank you for using the system.
```

Notes:

- The sample input/output is for demo purposes only. We'll use different test cases when grading.
- Some of the wordings can vary depending on how you implement the system. For example, if your superhero class is not called "MarvelHero" and/or its fields are named differently, your Debug Dump will be different.