The University of Texas at El Paso
Department of Computer Science
CS 3331 – Advanced Object-Oriented Programming
Instructor: Dr. Bhanukiran Gurijala
Spring 2025

Project Part 1

Academic Integrity Statement:
This work is to be done as a team. It is not permitted to share, reproduce, or alter any part of this assignment for any purpose. Students are not permitted to share code, upload this assignment online in any form, or view/receive/modify code written by anyone else. This assignment is part of an academic course at The University of Texas at El Paso and a grade will be assigned for the work produced individually by the student.

Instructions:
Your code must be written in Java. In the comment heading of your source code, you should write your name(s), date, course, instructor, project part 1, project description, and honesty statement. The honesty statement must state that you completed this work entirely on your own without any outside sources including peers, experts, online sources, or the like. Only assistance from the instructor, TA, or IA will be permitted. Generate Javadoc for your complete code.

Scenario:
You are tasked with designing a system to model and interact with space debris in Low Earth Orbit (LEO). Your system will allow scientists, space agencies, and policymakers to track debris, assess if debris is in orbit or out of orbit, and analyze its potential impact on space operations to raise awareness among decision-makers.

Part A:
Read the requirements described in Part B to complete Part A. Part A must be completed before implementing the requirements in Part B

1. Write a UML Use Case Diagram (Level II) for your system. With at least the following:
    a. 3 actors (e.g., Scientist, Space Agency, Policymaker)
    b. 3 Use Cases (e.g., Track Debris, Assess Orbit Status, Analyze Impact)

    c. 1 includes relationship
    d. 1 extends relationship
2. Write 2 use case scenarios based on Part B.
3. Write a UML Class Diagram to structure your code using the classes, requirements, and concepts described in Part B

Part B

1. Create the following classes (Note: some may be abstract)
    a. SpaceObject
    b. Debris
    c. Satellite
    d. MissionControl
    e. Tracking System
    f. ImpactAnalysis
    g. DebrisDensityAnalysis
    h. RunSimulation (where you have your main method)
    i. Any other additional Classes that you believe will be beneficial to help with successfully implementing this program
    j. Class considerations
        i. All classes should have appropriate methods and fields/attributes.
        ii. Use object-oriented principles such as encapsulation and inheritance.
2. Read files with debris and satellite information and store the information appropriately.
    a. Pick an appropriate data structure.
        i. Consider the time complexity.
        ii. Consider space complexity.
        iii. Ensure efficient retrieval and storage of objects.
    b. Consider the use of objects and how your objects will interact with each other.
3. Allow for user interaction.
    a. The system shall first capture the type of use interacting with the system. The allowed types of users include:
        i. Scientist
        ii. Space Agency Representative
        iii. Policymaker
        iv. Administrator
        v. Exit
    b. If the user is a Scientist, the following menu options shall be displayed:

        i. Track Objects in Space
        ii. Assess Orbit Status
        iii. Back: If the user selects this option, the system shall take the user to the previous menu screen (in this case, the system will present the user with options to select the type of user. E.g., it will present the menu option mentioned in 3a.)
    c. If the user is a Space Agency Representative, the following menu options shall be displayed:
        i. Analyze Long-term Impact
        ii. Generate Density Reports
        iii. Back: If the user selects this option, the system shall take the user to the previous menu screen (in this case, the system will present the user with options to select the type of user. E.g., it will present the menu option mentioned in 3a.)
    d. If the user is a Policymaker, the following menu options shall be displayed:
        i. Review Reports on Debris Impact
        ii. Assess Risk Levels for Future Space Missions
        iii. Back: If the user selects this option, the system shall take the user to the previous menu screen (in this case, the system will present the user with options to select the type of user. E.g., it will present the menu option mentioned in 3a.)
    e. If the user is an Administrator, the following menu options shall be displayed:
        i. Create User
        ii. Manage User
        iii. Delete User
        iv. Back: If the user selects this option, the system shall take the user to the previous menu screen (in this case, the system will present the user with options to select the type of user. E.g., it will present the menu option mentioned in 3a.)

4. In this part, your system should be able to handle the functionality for Scientist. The following functionality should be implemented:
    a. Track Objects in Space: On selecting this option, the Scientist should be presented with the following menu options:
        i. Rocket Body: If the user selects this option, the system shall provide a list of all Rocket Body with the following information:

      1. Record ID, Satellite Name, Country, Orbit Type, Launch Year, Launch Site, Longitude, Avg. Longitude, Geohash, and Days Old.
   ii. Debris: If the user selects this option, the system shall provide a list of all Debris with the following information:
      1. Record ID, Satellite Name, Country, Orbit Type, Launch Year, Launch Site, Longitude, Avg. Longitude, Geohash, and Days Old.
   iii. Payload: If the user selects this option, the system shall provide a list of all Payload with the following information:
      1. Record ID, Satellite Name, Country, Orbit Type, Launch Year, Launch Site, Longitude, Avg. Longitude, Geohash, and Days Old.
   iv. Unknown: If the user selects this option, the system shall provide a list of all Unknown with the following information:
      1. Record ID, Satellite Name, Country, Orbit Type, Launch Year, Launch Site, Longitude, Avg. Longitude, Geohash, and Days Old.
b. Assess Orbit Status: If the user selects this option, the system shall provide the following menu options:
   i. Track Objects in LEO: If the user selects this option, the system shall provide a list of all objects in the LEO with the following information:
      1. Record ID, Satellite Name, Country, Orbit Type, Launch Year, Launch Site, Longitude, Avg. Longitude, Geohash, and Days Old.
   ii. Assess if debris is still in orbit: If the user selects this option, the system shall perform the following functionality:
      1. To assess whether debris is still in orbit or has exited (decayed or deorbited), we can infer using the following key indicators:
         a. The debris is still in orbit if the approximate_orbit_type is defined (LEO, MEO, etc.), longitude has a valid value, days_old is < 15,000 days, and conjunction_count is >=1
         b. The debris has exited the orbit if the approximate_orbit_type is missing or unknown, longitude is missing or zero, days_old is > 15,000 days, and conjunction_count is 0 (not seen interacting recently).

            c. Calculate orbital drift (longitude deviation computed as abs(longitude – avg_longitude). If the orbital drift is greater than 50, classify the debris as “High Risk”. If the orbital drift is greater than 10, classify the debris as “Moderate Risk”, otherwise as “Low Risk”
        2. After assessing the orbit status,
            a. Save the computed/calculated information by creating new columns called still_in_orbit that can take the value of True/False, risk_level that can be High, Moderate, or Low as mentioned above. The new CSV should include all columns of the original CSV along with the assessed orbit status column.
            b. Write a new TXT file with the count of in-orbit vs exited debris. This TXT file should also include information about all the exited debris. The TXT file shall include at least the following information about the exited debris:
                i. Record ID, Satellite Name, Country, Orbit Type, Launch Year, Launch Site, Longitude, Avg. Longitude, Geohash, and Days Old.

5. Log all system interactions.
    a. Keep track of all operations performed, including timestamps
    b. Log entries should persist across sessions (append instead of overwriting)
    c. A sample of how a log file can look is provided below (not necessarily the only way – feel free to make it better):

```
“[2024-10-10 12:30:00] Scientist queried debris within 200 km of Satellite-456.”
“[2024-10-10 12:35:10] Debris-123 is still in orbit at an altitude of 400 km.”
“[2024-10-10 12:40:25] Debris density report generated: "High debris density detected
between 700-800 km altitude."
```

6. The user can exit the program by writing “EXIT” while in the main menu only. When the user exits the program
    a. Write a new (updated) Debris Tracking Report (similar to the original input, except with the new values computed as part of assessing if debris is still in orbit functionality)
    b. Save the latest log data.

7. Handle all exceptions appropriately.
    a. Ensure robust error handling for file operations, user inputs, and calculations.
8. Write Javadoc for your system.
9. Write a lab report describing your work (template provided)
    a. Any assumptions made should be precisely mentioned in the source code and described in the lab report.
    b. The lab report should contain sample screenshots of the program being run in different circumstances, including successful and failing changes.
10. Complete an individual code review on your code (template provided)
11. Schedule a demo with TA/IA for both check-in and final deliverables
12. **If the submission is past the deadline** Your report must have an additional section entitled "Why I submitted late". In that section, explain the reason why your submission was late. (Note: you will still be penalized the typical late penalty)

<u>Deadlines:</u>

<u>Check-in on April 13, 2025, by 11:59 pm:</u>

1. UML Class Diagram Progress (.pdf)
2. UML Use Case Diagram Progress (.pdf)
3. Current Progress Source Code (.java) – Commit current progress up to this point.

For each item (1-3)

a. Does not have to be complete. Ensure there are no compilation errors.
b. Should be a significant amount of work done (as determined by the instructional team).
c. Demo with TA/IA after 04/13. All the check-in demos should be completed by 04/17. The TA/IA will review for progress and provide informal feedback.
d. The TA/IA working with the team will announce their availability for check-ins on Blackboard.

<u>Final Deliverables on April 20, 2025, by 11:59 pm:</u>

1. UML Class Diagram (.pdf)
2. UML Use Case Diagram (.pdf)
3. Use case Scenarios (.pdf)
4. Source code (.java files)
5. Lab report (.pdf file)
6. Javadoc (entire doc folder)

7. Updated Debris Tracking Report (.csv)
8. Log (.txt)