**Introduction:**

The following project report reveals the way of developing a Student Management Information System (MIS) by Using Java; The key aim each of each of the subsequent projects is to build powerful console-base application and this is based on the principle of Object-Oriented Programming (OOP). The system also consists of an administration of student data where one can update student data, the student data can also be viewed and if needed the data of the students can be deleted. Some of OOP concepts that is employed in the project include inheritance, polymorphism, encapsulation and abstraction that enhance the modularity of the project with easy maintenance. In addition, file I/O is employed to deal with the storage of data, as for the exception handling to enhance the stability of the system and the usability of the application. The project also provides an example of how one can develop a viable MIS with Java, since educational institutions’ and the like’s needs will not be left unfulfilled. Through this system, the users will be in a position to have many ways in which they can be able to search through the student information hence, the reason why good structure of software is good for construction of good and efficient application. This document seeks to establish the following; The details of the project undertaken The methodology involved in undertaking the project The choice of programming methodologies used.

**Features and Functionality**

The MIS project includes the following key features and functionalities: The MIS project includes the following key features and functionalities:

**Menu-Based Interface:**

Application offers the user interface where the users can have several menus presenting them the possibilities of working with the student data. The menu options include:

Add Information

Update Information

View Inventory

Delete Information

Exit

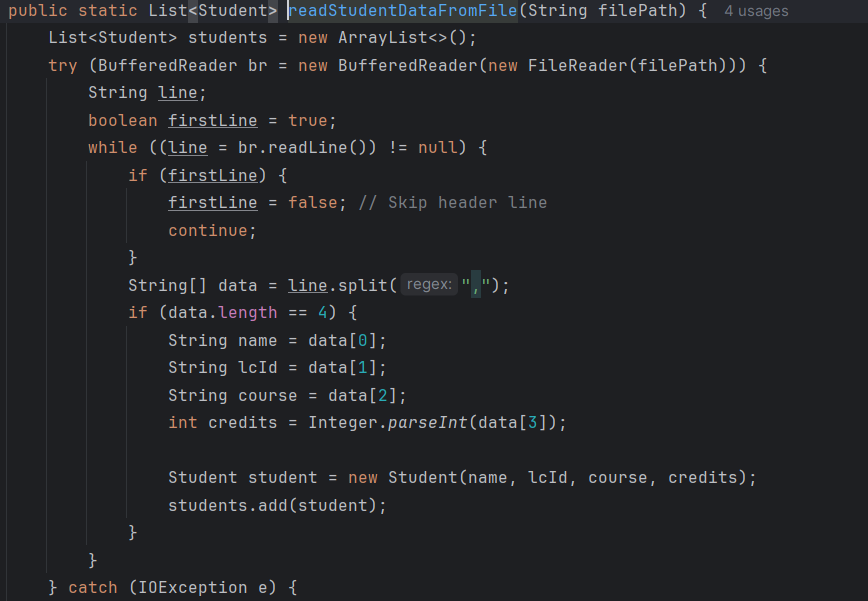
Student Data Management:

The system lets the users create, read, update and delete data belonging to students. Every operation is supported with proper data validation and error handling in order to maintain soundness of the application.

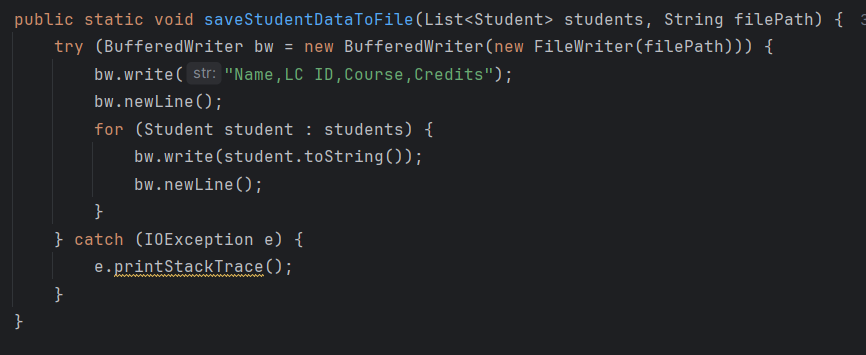
**File Handling:**

Student data baseline is maintained in the text file on permanently. This file is used by the system to read and write to so that all information concerning students is stored between sessions.

1. **File Reader:**

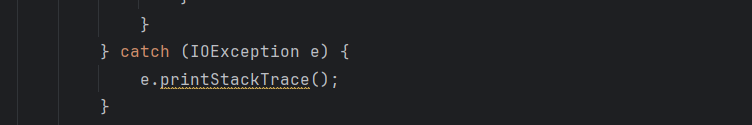


1. **File Writer:**



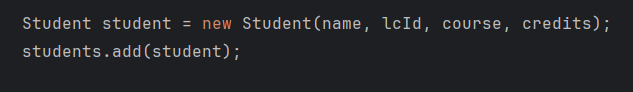
**Exception Handling:**

Apart from that, there is also extended DIT and AT containing elaborate exception control in order to handle problems like accessing files, wrong input or any other unforeseeable difficulty. This makes the application as stable as possible, as well as offer the user proper feedback indications.

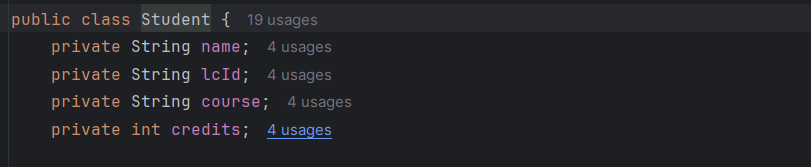


**OOP Concepts:**

The project is designed using core OOP concepts: The project is designed using core OOP concepts:



**Encapsulation:** The operations on data and their respective information are wrapped within the Student class giving a proper structure and preventing direct access to the students’ data.



**Inheritance and Polymorphism:** While not very intentionally invoked in this project, the concepts of inheritance and polymorphism are considered for their possible future expansion of the system.

**Abstraction:** With this system, all the high level file handling is transparent and what one finds is that there is a simple method call to access a student’s data.

**Implementation Details**

Details of the MIS system implementation with regard to development, a lot of concentration was laid on features such as code quality, readability and maintainability. Key implementation details are as follows: Key implementation details are as follows:

**Class Design:**

**Student Class:** Represents individual students, having fields that include name, LCID, course, and credits. This class also contains getter and setter for these attributes, thus” the attributes are well encapsulated.

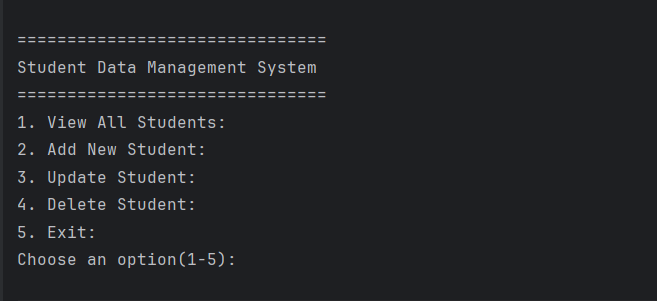
**StudentDataManagement Class**: Is responsible for the processes of data storage and processing; adding, modifying, visualizing, and erasing student information. This class also deals with user input and control of the menu as well.

**Data Persistence:**

Record keeping of the student details is done in a text file in this system. The file is open each time the application starts and every time the data in the application is modified. This helps in avoiding cases whereby data has been input in a particular session is not retrieved in subsequent session.

**User Interaction:**

It also offers the various options which a user, in this case, anyone involved in a case, is likely to make to complete his work. Messages include information to help the end user correct an error he/she might have made when entering information.



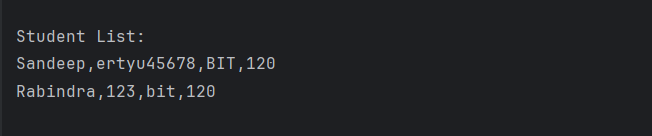
**Testing:**

A lot of testing was conducted in order to make sure the system performs according to expectation. The system was checked with different inputs that include boundary cases in order to ensure that all available functions work to the expected level and to also ensure that the system handles exceptions in the right way.

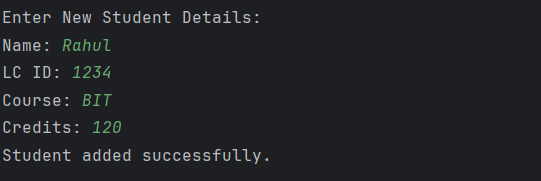
**Challenges and Solutions**

During the development of the MIS system, several challenges were encountered: During the development of the MIS system, several challenges were encountered:

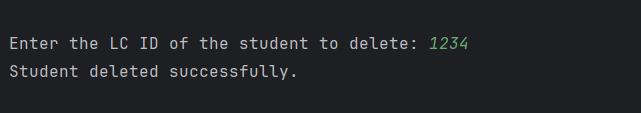
**View all Students:**



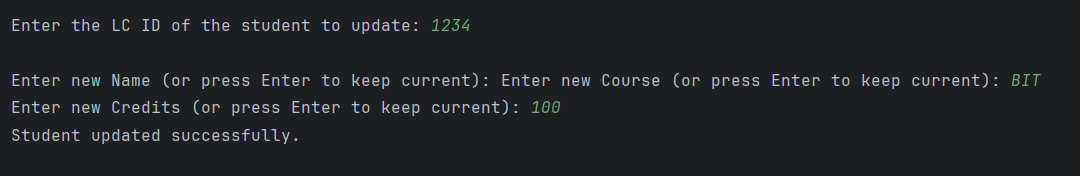
**Add Student:**

****

**Delete Student:**

****

**Update Student:**

****

**Conclusion:**

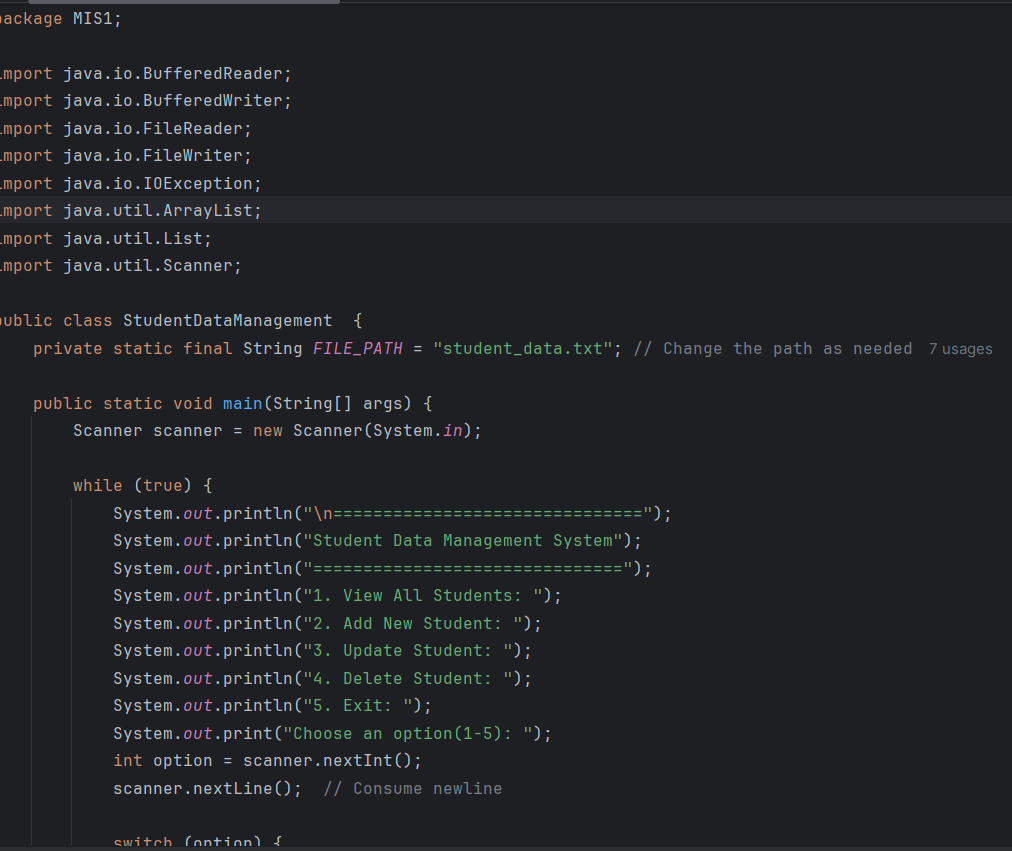
The Management Information System (MIS) project as shown how Object-Oriented Programming principles can be applied in Java. Here are basic set of features of the system: This Data Management System basically focuses on storing student records making use of features such as: create, modify, search, and remove entry among others. The project also focus on file managing, exception managing and standard of the code. All in all the MIS project is an excellent example of how to design a console based application in java while following standard procedure of software development.

**References:**

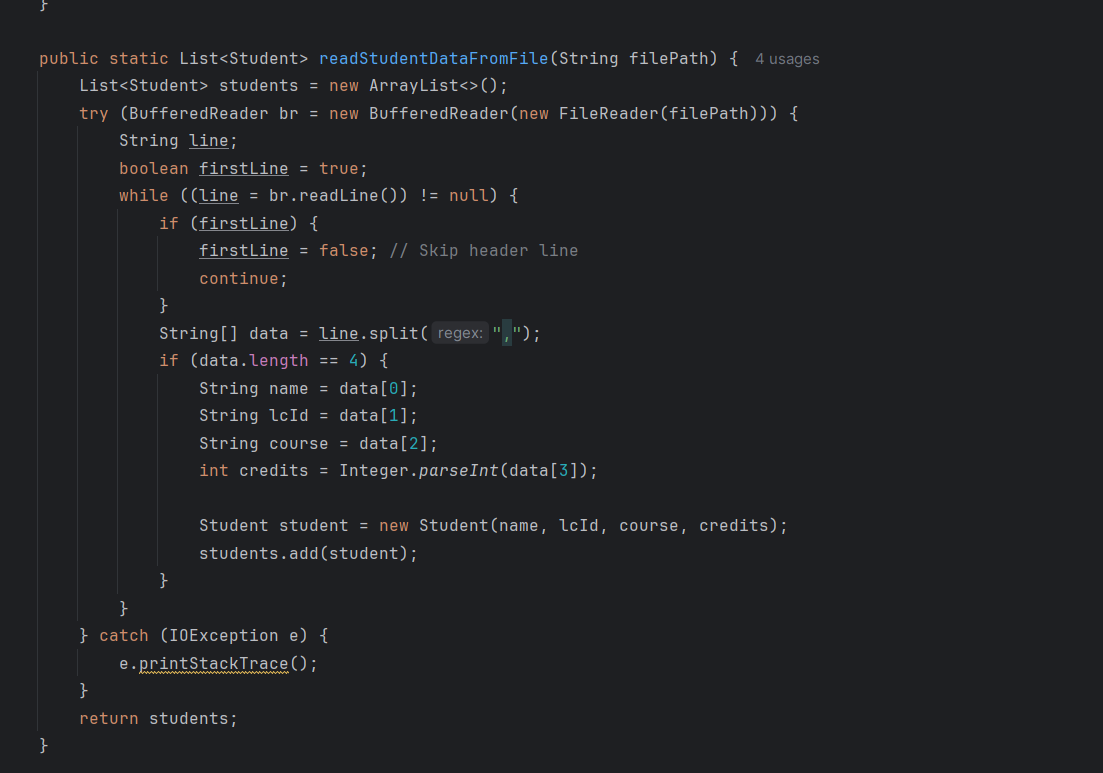
* Balagurusamy, E., 2019. Programming with Java: A Primer. 6th ed. New Delhi: McGraw Hill Education.
* Bloch, J., 2018. Effective Java. 3rd ed. Boston: Addison-Wesley.
* Deitel, P.J. and Deitel, H.M., 2017. Java: How to Program. 11th ed. Upper Saddle River, NJ: Pearson.

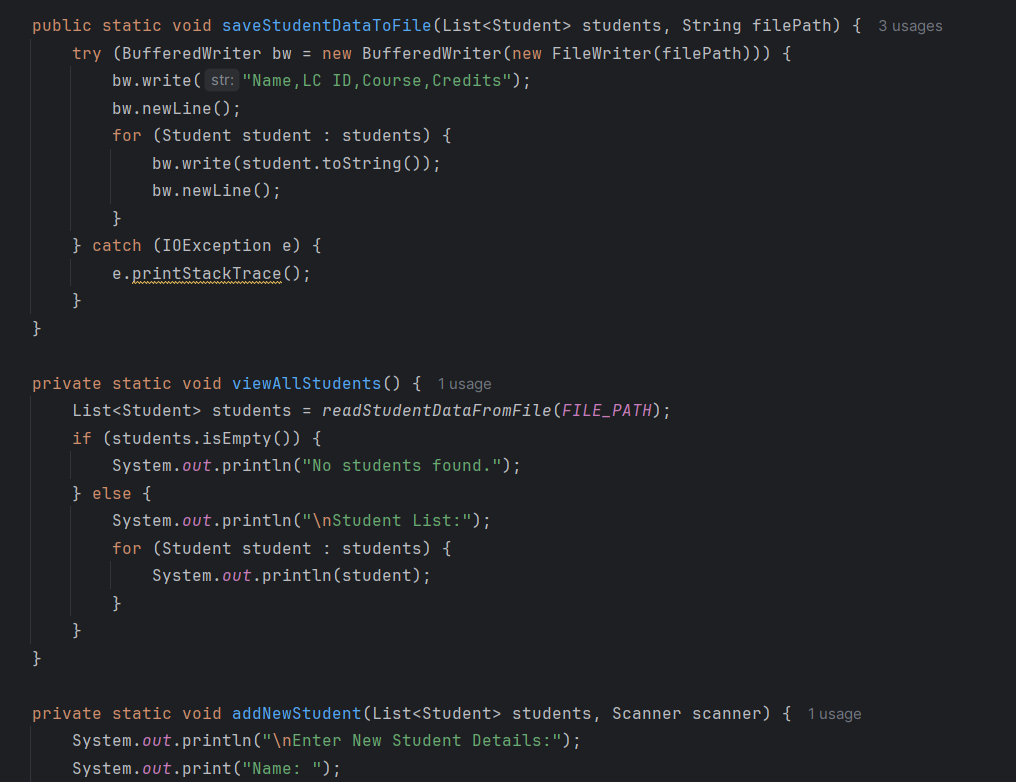
**Appendix:**

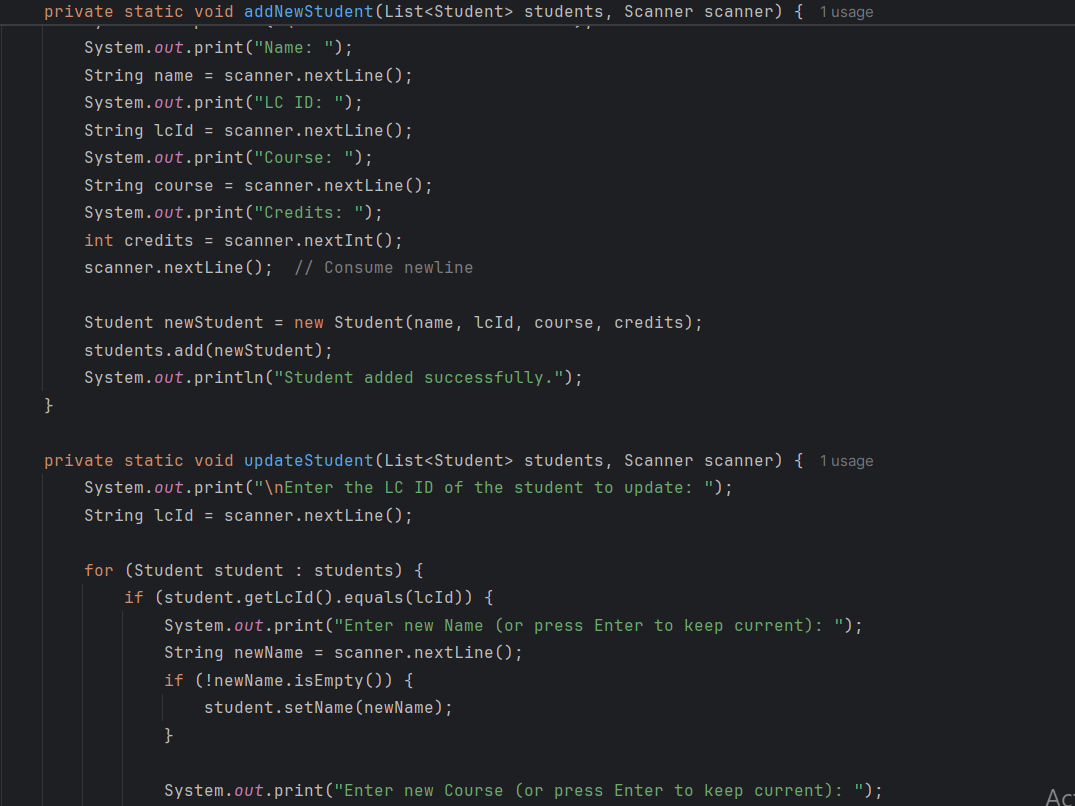
StudentDatamanagement:

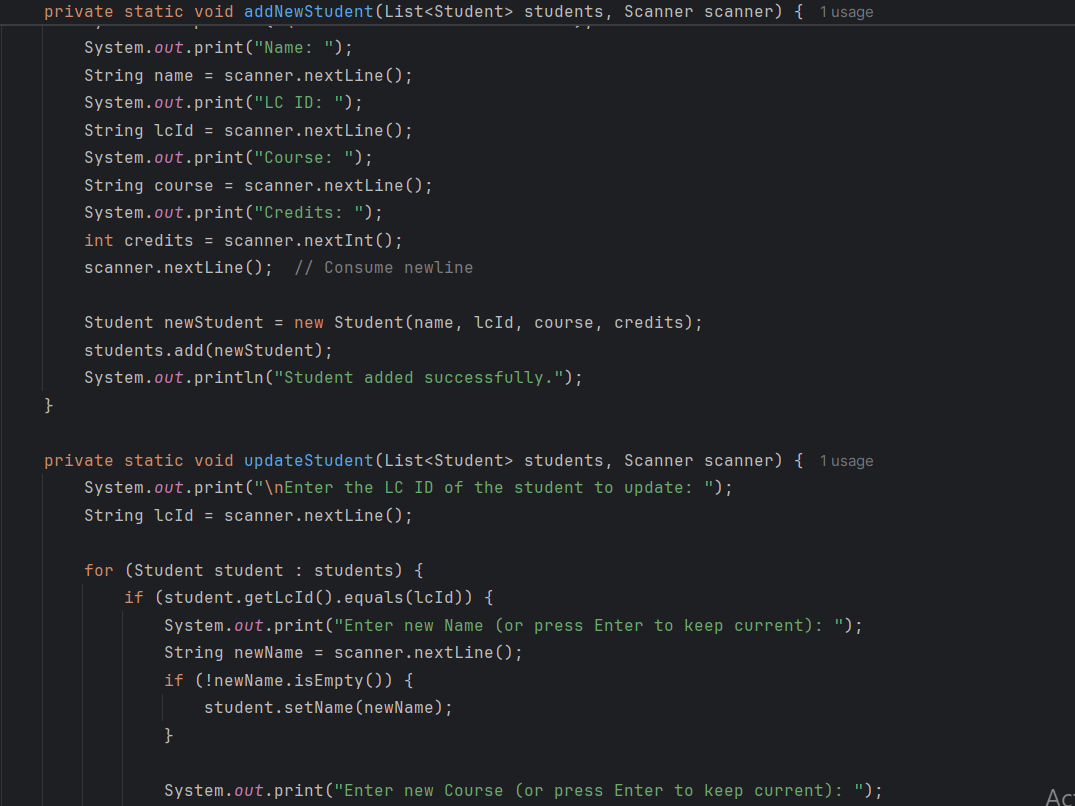
****

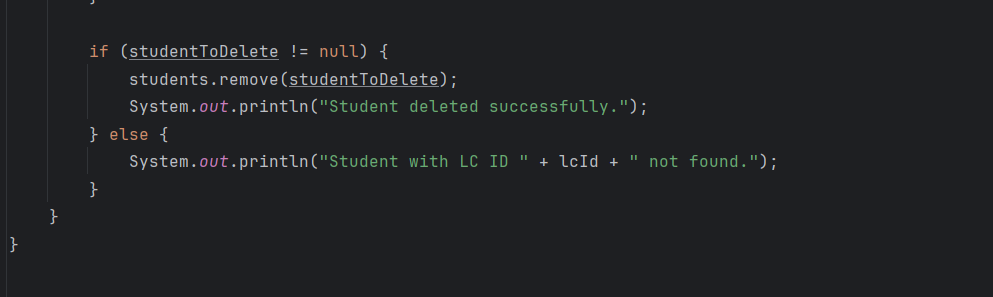
****

****

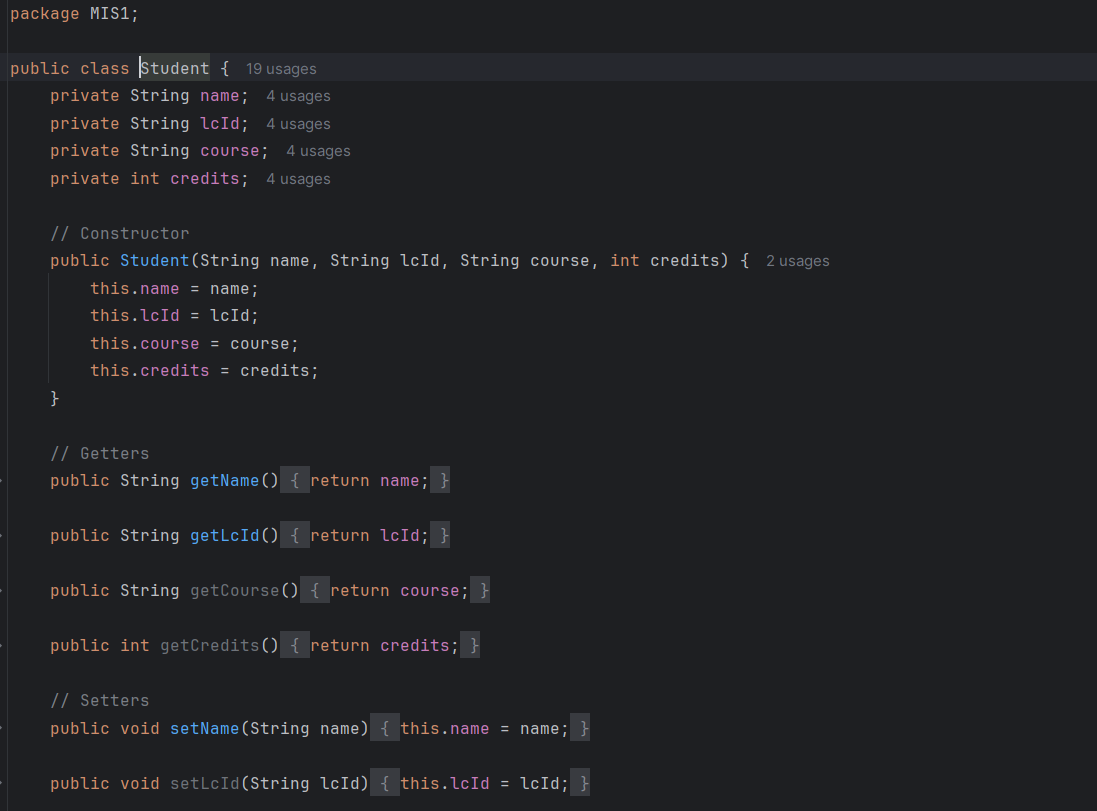
****

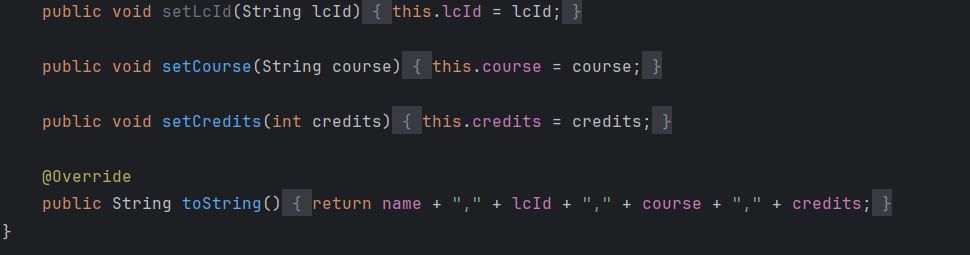
****

****

****

Student.java

****

****