Assignment1: COSC1200

**Instructions:**

1. Complete all tasks provided below.
2. Ensure your code is properly commented and follows Java naming conventions.
3. Use appropriate data types for variables.
4. Submit your solutions in a single .zip file in DC connect

**Task 1: Basic Class and Object Creation (4)**

Create a class called Student with the following:

* **Fields:**
  + name (String)
  + age (int)
  + studentID (String)
* **Constructor:**
  + A constructor that initializes all the fields.
  + A default constructor.
* **Methods:**
  + displayDetails(): Prints the student’s details in a user-friendly format.

Create a main method to:

1. Instantiate two Student objects.
2. Call displayDetails() on both objects.

**Task 2: Encapsulation and Getters/Setters (4)**

Update the Student class to:

1. Make all fields private.
2. Add public getter and setter methods for each field.
3. Modify the displayDetails() method to use the getter methods instead of directly accessing the fields.

Test the getters and setters in the main method by:

* Changing the values of the fields for one of the Student objects.
* Printing the updated details.

**Task 3: Static Members (3)**

Enhance the Student class by:

1. Adding a static field universityName (String) with a default value (e.g., "Tech University").
2. Adding a static method changeUniversityName(String newName) to update the university name.

In the main method:

* Use the static method to change the university name.
* Verify that the change is reflected for all Student objects.

**Task 4: Code Styling and Naming Conventions (4)**

Ensure that your code follows these guidelines:

1. **Class Naming:** Classes should be named using PascalCase (e.g., Student, CourseDetails).
2. **Variable and Method Naming:** Use camelCase for variables and methods (e.g., studentID, displayDetails).
3. **Constants:** Declare constants using final and name them in uppercase with underscores (e.g., MAX\_AGE).
4. **Indentation:** Use consistent indentation (4 spaces per level).
5. **Comments:** Add comments to explain the purpose of classes, methods, and complex logic.
6. **Appropriate Data Types:** Use suitable data types for variables (e.g., int for age, String for names).