Student Management

Team Members Names & Ids:

Marwan Mostafa (Lead): 23-101165.

Yousef Walid : 23-101117.

Abdelrahman Ahmed: 23-101010.

Abdelrahman Amr: 23-101225.

Saif El din: 23-101113.

Application Description:

The Educational Management System (EMS) is designed to manage information for an educational institution. This includes students, instructors, and courses. The system allows administrators to add, remove, and display information about these entities. It also provides functionalities for instructors to manage their courses and for students to manage their enrolments and view their grades. The system is implemented using object-oriented programming principles and provides a menu-driven interface for user interaction.

Program Structure (Description of each class):

- **Administrator**
  - This class represents the system administrator and has the following functionalities:
    - Manages students (add, remove, display information)
    - Manages instructors (add, remove, display information)
    - Manages courses (add, remove, display information)
    - Handles user menus (administrator menu)
- **Student**
  - This class represents a student in the system and has the following functionalities:
    - Registers for courses
    - Drops courses
    - Views grades in a specific course
    - Gets statistics (average, minimum, maximum) for all courses enrolled in
    - Handles user menus (student menu)

- **Instructor**
  - This class represents an instructor in the system and has the following functionalities:
    - Adds courses to teach
    - Removes courses from teaching list
    - Sets grades for students in a course
    - Performs statistics (maximum, minimum, average) on a course
    - Handles user menus (instructor menu)
- **Course**
  - This class represents a course in the system and has the following functionalities"
    - Add/Remove instructor ids that teach this course
    - Add/Remove student ids and grades that are enrolled in this course
    - Get statistics (maximum, minimum, average) on all the grades in this course

Explanation of The OOP Applied in The Project:

- **Inheritance:** The classes Student and Instructor inherit from the User class, which likely contains common attributes and functionalities for all users of the system (username, password, etc.).
- **Encapsulation:** Data members of each class are likely private, and member functions provide access and manipulation of this data.
- **Polymorphism:** The handleMenu() function is implemented in each class (Administrator, Student, Instructor) with specific functionalities relevant to the user type.

Workload Distribution Within the Team:

Yousef Walid: carried out core functionalities in the Course class

Saif Alaa El din: implemented the student enrollment functionalities for the Student class and overloaded the = operator.

Abdelrahman Amr: contributed to the development of the Instructor class and overloaded the = operator in both the Instructor and Course classes.

Abdelrahman Ahmed: performed main operations in the Administrator class.

Marwan Mostafa (Lead): Developed the UML and the remaining functionalities and oversaw system integration in addition to Input validation, menus, and dynamic memory management.

Test Cases (with input and output screens):

A non-integer input is entered:

```
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
not-a-number
Invalid input. Please enter a number: 1
Student username:
```

An not-in-choices integer is entered:

```
Username: admin
Password: Admin123!
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
9
Invalid choice! Please try again.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
```

When a course is created with a duplicate code:

```
Course name: Calculus 2
Course code: C-MA112
Course credits: 3
Calculus 2 is added successfully.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
5
Course name: Calculus 3
Course code: C-MA112
Course credits: 4
A course with the same code is found. Try again with different code.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
```

When an invalid course code is entered:

```
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
5
Course name: Programming Fundamentals
Course code: C--CS
Course credits: 3
Course code is invalid. It must be all uppercase letters with at least one number and one dash.
Course can not be created due to invalid information.
Failed to add a course with invalid information.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
```

When a user (instructor or student) is created with a duplicate username:

```
Student username: marwan
Student password: Marwan123!
marwan is added successfully.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
1
Student username: marwan
Student password: Marwan123!
A student with the same username is found. Try again with different username.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
```

When an invalid username is entered:

```
Username: admin
Password: Admin123!
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
1
Student username: Abc%
Student password: Ahmed123!
Usernames must be 4-20 characters long and can contain letters (a-z), numbe
rs (0-9), and periods (.). Usernames cannot contain an ampersand (&), equal
s sign (=), underscore (_), apostrophe ('), dash (-), plus sign (+), comma
(,), brackets (<,>), or more than one period (.)
Failed to add a student with invalid credentials.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
```

When an invalid password is entered:

```
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
3
Instructor username: ahmed_gaber
Instructor password: Ahmed123
Password is invalid. It must be 8-32 characters long and contain at least one
uppercase letter, one lowercase letter, one digit, and one special character.
Failed to add an instructor with invalid credentials.
1. Add Student
2. Remove Student
3. Add Instructor
4. Remove Instructor
5. Add Course
6. Remove Course
7. Display All Information
8. Logout
```

When trying to register the same course (same with the instructor)

```
1. Enroll in a new Course
2. Drop a Course
3. View Grade in a Specific Course
4. Get statistics in all courses
5. Display All Information
6. Logout
1
2
Available courses to register in:
Programming Fundamentals(C-CS112), Algorithms(C-CS114)
Type the course code: C-CS114
Algorithms is registered successfully.
1. Enroll in a new Course
2. Drop a Course
3. View Grade in a Specific Course
4. Get statistics in all courses
5. Display All Information
6. Logout
1
2
Available courses to register in:
Programming Fundamentals(C-CS112), Algorithms(C-CS114)<-- Already registered
Type the course code: C-CS114
Course with that code already registered
1. Enroll in a new Course
2. Drop a Course
3. View Grade in a Specific Course
4. Get statistics in all courses
5. Display All Information
6. Logout
```

List of References:

https://www.geeksforgeeks.org/copy-constructor-vs-assignment-operator-in-c/
https://stackoverflow.com/questions/625799/resolve-build-errors-due-to-circular-dependency-amongst-classes
https://stackoverflow.com/questions/672373/can-i-call-a-base-classs-virtual-function-if-im-overriding-it#fromHistory