

# SE 116 Introduction to Programming II

Lab No:	01
Topic:	SE 115 RECAP

## TASK 1:

You are responsible for developing a **small simulation** of **OASIS** (the Student Information System of IUE).

(OASIS will **not** be a separate class; it will be your main class. Act accordingly.)

In **OASIS**, there will be **students** and **instructors**. Each student has a corresponding `id`, `name`, `age`, and `GPA`. You will determine the data types of these class fields.

In the **Student** class, there will be a method to display the corresponding **Student** object's information, along with **accessor (getter) and mutator (setter) methods** to set and get the class fields. Implement a parameterized constructor for creating objects from this class.

Additionally, each **Instructor** has a corresponding `id`, `name`, `age`, and a list of `students` they are responsible for.

- This list's **size is fixed at 3**, and you are **not allowed** to use collection classes such as `ArrayList`, `LinkedList`, etc.
- Implement a parameterized constructor for creating objects from this class, initialize the list of students in that constructor.

The instructors should be able to add and remove Students into their Student Lists with the following methods;

```
addStudent(Student student);
```

```
removeStudent(int id); : Be careful about the removed slot of the array what happens to it?
```

Also the instructor should be able to find the requested Student by its `id` and change his/her GPA with the following methods:

```
findStudent(String id);
```

```
changeGPA(Student student, double newGPA);
```

You will use the 2 methods above in a single method below, they are helper methods to this method;

```
findStudentAndChangeGPA(String id, double newGPA);
```

Finally, the instructor should be able to see his/her list of students he/she've to be dealt with the following method:

```
displayAllStudents();
```

Create some Student and Instructor objects and try each functionality one by one.

## TASK 2:

In math, the natural “log” or “ln x” is the inverse of “e<sup>x</sup>”. There are series that help you to calculate “e<sup>x</sup>” and “ln x”.

Through sum of reciprocal of factorials “e<sup>x</sup>” can be calculated like this:

$$\sum_{i=1}^k \frac{x^i}{i!}$$

**HINT:** You will need an additional helper function to calculate factorial of i.

Through Exponential and logarithm series you can calculate “ln x” like:

$$\sum_{n=1}^k \frac{(-1)^{n-1}(x-1)^n}{n}$$

**Note:** These series approximate the answer as they go towards infinity. We will limit “k” to a value that is entered by user. Therefore, calculation will stop at that value.

Write a **JAVA program** which calculates “ln x” or “e” depending on the user’s wish, with 2 **functions**. The name for the functions is like below:

```
calcEStep (int k, int x);  
calcLnStep (int k, int x);
```

You cannot change or add parameters to these functions. Also, you don’t need to use any extra functions. In your main function take “k” and “x” values from user and call the appropriate functions to calculate “e<sup>x</sup>” second to calculate “ln x”.