# TP1

## Exercise 1: Bank Account Management System

Develop a class named BankAccount, which models a bank account with an account number, an account holder's name, and a balance.

Class: BankAccount

Attributes:

- `accountNumber`: A unique integer that represents the bank account number.
- `accountHolderName`: A string that holds the name of the account holder.
- `balance`: An integer to represent the account balance.

Methods:

- `getAccountNumber()`: Returns the account number as an integer.

- `getAccountHolderName()`: Returns the account holder's name as a string.

- `getBalance()`:  Returns the balance as an integer.

- `deposit(int amount)`: Accepts an amount to deposit into the bank account and updates the balance (returns void).

- `withdraw(int amount)`: Accepts an amount to withdraw from the bank account, should check if the balance is sufficient, and update the balance. Returns void.

## Exercise 2: University Management System

Develop classes named 'Student', 'Course', and 'Instructor' that model a university management system.

Classes

1. `Student`

Attributes:

- `studentId`: A unique integer that represents the student ID.
- `firstName`: A string that represents the student's first name.
- `lastName`: A string that represents the student's last name.
- `courses`: An ArrayList that stores the courses in which the student is enrolled.

Methods:

- `getStudentId()`: Returns the student ID as an integer.
- `getFirstName()`: Returns the student's first name as a string.
- `getLastName()`: Returns the student's last name as a string.
- `getCourses()`: Returns a list of courses in which the student is enrolled.
- `enroll(Course course)`: Adds the course to the student's list of courses (returns void).

2. `Course`

Attributes:

- `courseId`: An unique integer that represents the course ID.
- `courseName`: A string that represents the name of the course.
- `instructor`: An Instructor object which represents the Instructor of the course.

Methods:

- `getCourseId()`: Returns the course ID.
- `getCourseName()`: Returns the course name.
- `getInstructor()`: Returns the Instructor of the course.

3. `Instructor`

Attributes:

- `instructorId`: A unique integer that represents the Instructor ID.
- `firstName`: A string that represents the instructor's first name.
- `lastName`: A string that represents the instructor's last name.

Methods:

- `getInstructorId()`: Returns the instructor ID.
- `getFirstName()`: Returns the instructor's first name.
- `getLastName()`: Returns the instructor's last name.

## Exercise 3: Custom ArrayList Implementation

Develop a class named 'CustomArrayList' that will mimic the basic functionality of the Java ArrayList.

Class: CustomArrayList

Constructors:

- `CustomArrayList()`: A default constructor that creates a CustomArrayList with an initial size of 10.
- `CustomArrayList(int initialSize)`: A constructor that allows setting an initial size for the ArrayList.

Methods:

- `void add(Object obj)`: A method that places an object at the end of the CustomArrayList.
- `void add(int index, Object x)`: A method that places an object at a specific position in the CustomArrayList.
- `Object get(int index)`: Returns the object from a given position in the CustomArrayList.
- `int size()`: Returns the number of objects currently in the CustomArrayList.

- `boolean isEmpty()`: Returns whether the CustomArrayList is empty.

- `boolean isIn(Object x)`: Checks whether a particular object exists in the CustomArrayList.

- `int find(Object x)`: Returns the position of the first occurrence of an object starting from position 0, if not found, should return -1.

- `void remove(Object x)`: Removes the first occurrence of an object from the CustomArrayList.