# CS201 Project 2

Binary Search Tree for Fast Searching

## General Description

The aim of this project is to create Binary Search Trees (BST) for fast searching using the data retrieved by a database system. The database system contains only one table that stores information of students. A student object has the following attributes:

| Attribute | ID | Name | Surname | Age | GPA |
|---|---|---|---|---|---|
| Type | Integer | String | String | Integer | Real |

ID is a unique integer with 6 digits, like your student ID. Name and surname are strings and their combination is unique. Age is an integer in the range [10, 100]. GPA is a real value that is in the range [0, 4].

Your job in a nutshell is to read data from a database system (in our case it is only a simple csv file), then form a binary search tree for ID and the combination of Name and Surname attributes (So, in total 2 BSTs), create data manipulation methods in each BST, and finally create exact and interval search methods inside each BST.

In more detail, for each week you should do:

## Weekly Task

### Week 1: Forming Student Class and Binary Search Trees

For this week, you should implement a Student Class with the attributes mentioned in the above table. You should get all the attributes as input to your constructor method.
Then, for ID and the combination of Name and Surname, you should create a BST such that the node of the BST has the same type as the attribute. So, in total, there should be 2 BSTs, one for ID with type Integer, and the other for combination of Name and Surname with type String.

### Week 2: Modifying Data

For this week, you should read data from a given large file line by line, and for each line, you should create an object from the Student class. Then, you should insert the new student object to its proper position inside each BST. For example, if you read the following in a line:

123456, Mehmet Arda, Eren, 24, 3.77

First, you need to create a Student object with such attributes. Then, insert a node with key “Mehmet Arda Eren” to the BST corresponding to Name and Surname, and another node with key 123456 to the BST corresponding to ID. You should assume that ID and combination of Name and Surname are unique.

The large file will be uploaded and given to you by us.

Also, you need to implement a delete method for each BST too. For example, if delete method for value “Mehmet Arda Eren” is called inside the BST corresponding to Name and Surname attributes, it should delete the node that has the key “Mehmet Arda Eren”.

An example of the data inside the file is:

1, 123456, Mehmet Arda, Eren, 24, 3.77
2, 789012, Arash, Mehrabi, 28, 1.93
3, 112233, Yiğit, Demirşan, 23, 4.00
4, 824901, Mohammad Ali, Foroughi Zokaolmolk, 35, 2.14

where the value before the first comma is the row number and the rest are student’s attributes with the following order: ID, Name, Surname, Age, GPA.

## Week 3: Search

For this week, you should implement two search methods for each BST. One exact search method and one interval search method. For the exact search method, you should implement a method for returning a node that its key is equal to a particular value. For example, the exact search method inside the BST corresponding to ID can take 123456 integer as its argument, and either return the student whose ID is equal to the given integer, or return null meaning it couldn’t find such student.
In the interval search, it can be asked to return all the students whose names are bigger than “Arash Mehrabi” and lesser than “Can Ethem Yildiz” in the BST corresponding to Name and Surname attributes. You should implement interval search for both BSTs.

# Project Structure

- You will use the Java programming language for this project.
- At first, please edit your github handle to be your name and surname so that we can identify you. If your github handle is “fullmetal-programmer”, it’s no good for us. If your name is “Arash Mehrabi”, please change it to something like “arash-mehrabi” or anything that can be identifiable. Also, please change your github profile name to your name and surname too. Then, use the following GitHub assignment, and create a repository for this project: https://classroom.github.com/a/MefcBfW0

# Important Notes

- You must come to the lab sessions to be graded for your work.

- You MUST accept the GitHub assignment with your name and surname only. Those who cannot be verified by us will receive 0.
- External library usage is forbidden.
- **The code can only be written and be submitted to the github during the lab hours.** However, you can practice and research outside of the lab hours.
-

If you have further questions, don't hesitate to contact me by my email at arash.mehrabi@ozu.edu.tr.
Good luck and Başarılar!