Project 2: Hashing

## Project Deadline: **Saturday 17 May 2014**

## Number of Students/Project:

## **Two** (from same section group)

* **Three** (from the same section group).

 **If you choose to have 3 members in the group then you have to implement the hashing in buckets.**

**Basis Considerations**

**// Start Cheating Rule ///////////////////////////////////**

## If (Cheated (anyStudent) == True)

## {

## SetPoints (anyStudent, 0);

## SetPoints (anyStudent.groupMembers, 0);

## }

**// End Cheating Rule /////////////////////////////////**

# Grading

50% on testing the functionalities

50% on coding

# Develop the hashing indexing for a data file.

Develop an application that proposes to the user 3 options:

Option 1: Add a new student

Option 2: Search for a student by ID

Option 3: Delete a student by ID

Display all the students (**bonus**)

## Option 1:

The user should enter the following information for the student:

* ID (integer composed of 8 digits, example 20101234)
* Full Name (a string of maximum 30 characters)
* Birth date
* GPA

Your application should display the address in the data file where the record should be stored and the address where it is stored (in case of collision).

## Option 2:

The user should enter and ID.

You will display to the user the address where the record should exist. If the record exists then display its address and the record data.

## Option 3:

The user should enter the ID.

You will display to the user the address where the record should exist. If the record exists, then display its address and delete the record.

## Option 4:

You should display all the records in the data file without displaying empty records.

General direction:

* You are free to create any hashing function that is not “all synonyms”
* Your application should not crash at all during test
* You must validate the input data (student info). You should make sure that all the data are entered. Make sure that the GPA is a numeric value, etc.