# Before starting in the main menu

Reading the students file and sort it in dictionary, the sorting included the main dictionary and the IDs of the students as keys, and inside the dictionary included 4 values which is:

* student’s name
* student’s ID
* student’s major
* list of the student’s courses

And inside the list of courses there is more lists, and each list include the info of the course as:

* Course code
* Course name
* Credit hours
* Grade

And the other important thing in this sorting was removing the students with ID number less or more than 9 digits, and removing students with IDs start with number other than 2 and the IDs that ends with number other than 0, and these students weren’t included in the dictionary.

After sorting the file information in the dictionary, we start the work in the main menu.

# The main menu

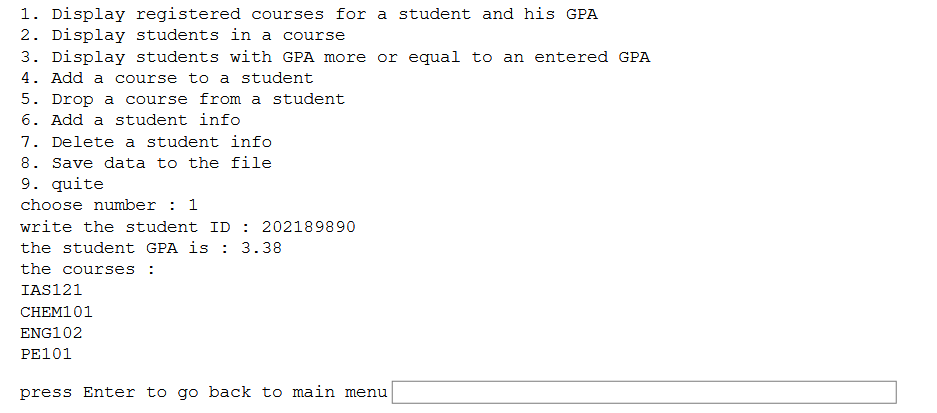
Before coding the options, we used if statement to lead the user to the right function by knowing the entered number and the after printing the options with each associated number, and by using classes to do the options.

## Option 1

The first option was (Display registered courses for a student and his GPA.)

The first thing we did was finding if the student in the dictionary or not by prompting the user for the student ID, the search in the dictionary if the student ID is there, and if it’s there, we search if the student has courses in the dictionary, and if he has courses we calculate the overall GPA by using loops and identify the course credit hours and grade, so we used if statement to divide the grades and put the associated weigh to each grade, lastly we print the student’s courses and GPA, and this option has some exceptions and error messages which is :

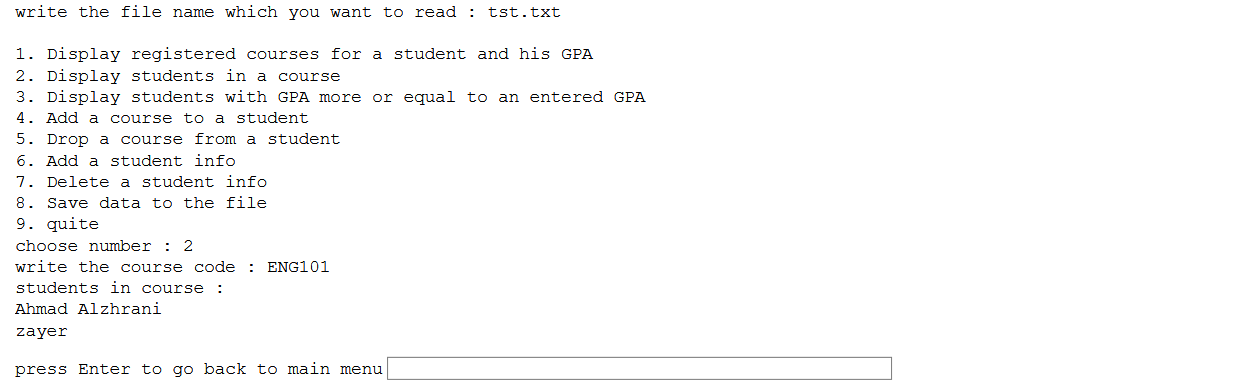
* If the ID in the list or not
* If the student has courses or not



## Option 2

Option 2 was (Display students in a course)

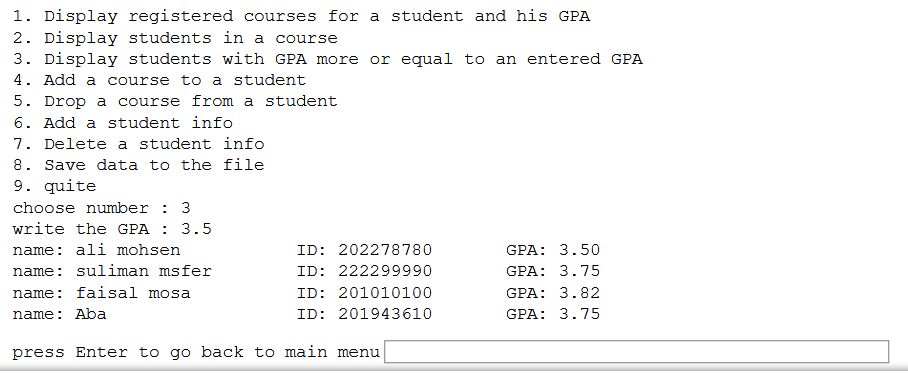
First we asked the user to enter a course code to search for it, and then we print a message to clarify the output that says (students in course: ), then we used a for loop to check for every id in our dictionary, then another for loop to check for every course a student have. If the course was found we increase the variable (coursenum) by one as a counter, and then print the student’s name that has the course. We used an if statement to check if no one has the course. If the counter we used earlier equals zero we print “No student has this course”.



## Option 3

Option 3 was (Display students with GPA more than or equal to an entered GPA value)

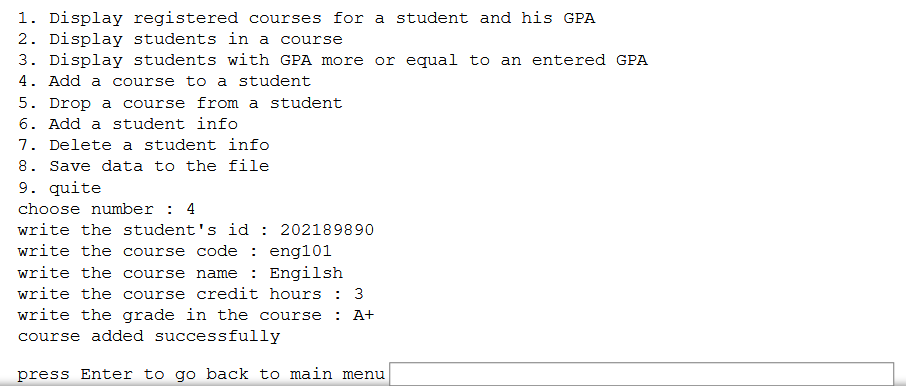
The first thing we did was asking the user for the GPA to search for the student that their GPA is higher than the entered, after that we check if the entered GPA is appropriate or not, then we called the class and the appointed function which is aboveGpa(), at the first we use loop to move in the students lists of course and there is another loop inside it to calculate the GPA for each student (same as option 1) but before that we check for each student if there is courses in their list (because if we used the GPA calculator loop on empty list that will rise an error) and we use if statement to find the GPA that more than the entered GPA and if it is then append it to new list with the student’s name and ID then print it in the main function.



## Option 4

This option is (Add a course to a student)

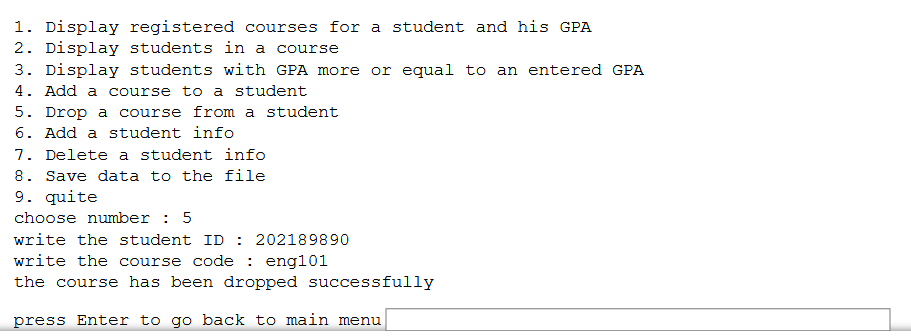
And it is pretty simple because we can add course without checking if it is there already, firstly we ask the user to write the ID of the student to add course to him then check if the ID in the dictionary then if it is there we ask the user to write the code and the name and credit hours of the course and his grade in the course. Then add these 4 to new list and add this list to the list of courses of the student, and we put additional exception if the entered credit hours are not integer.



## Option 5

The option is (Drop a course from a student)

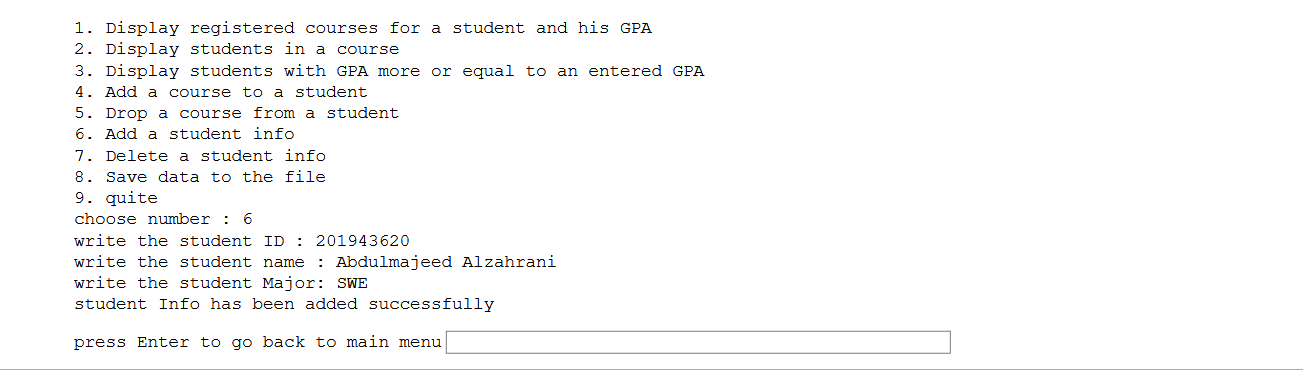
As always we started with asking the user for the student’s ID the check if it is in the dictionary or not, then and also check if student have courses or not, then ask the user for the code of the course he want to drop, after that we check if the student has this course or not by using loop and after we find the course we remove the course list from the courses list of the student and show the success message.



## Option 6

Option 6 is (Add a student info)

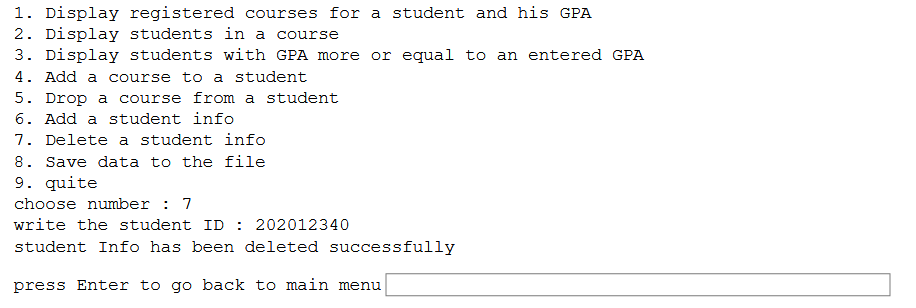
First, we asked the user for student id, and then we check the validity of that id by using if statement. if the id is less or more than 9 digits or if the ID start with number other than 2 or ends with number other than 0. If id is not valid we print an error message. And then we check if the id already exist, if it is we print an error message. If the id is valid and it is not already in the list we ask for the other info (Name, Major) and then we append the student info in the list and print a success message.



## Option 7

Option 7 is (Delete a student info)

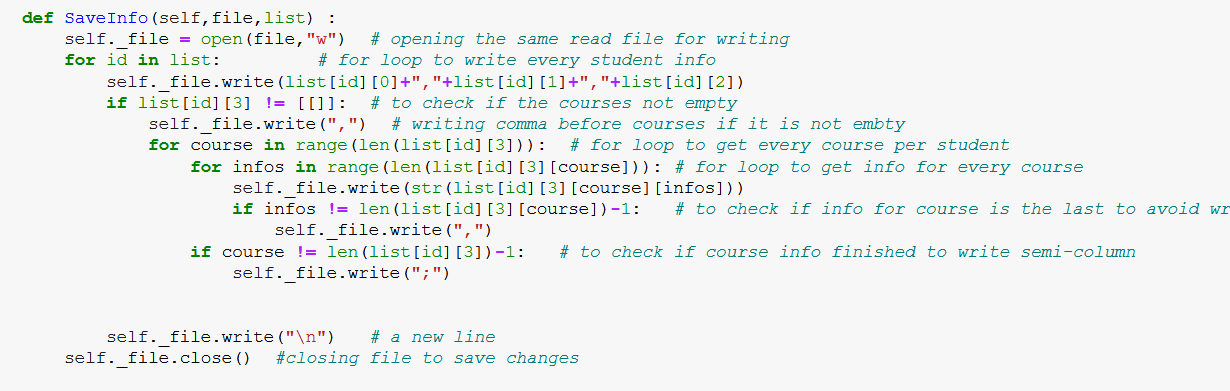
First, we asked the user for the id he want to delete from the list, then we used if statement to check if the id is in the list or not. if it is in the list we delete it using pop function. If it’s not we print an error message.



## Option 8

Option 8 is (Save data to the file)

In this option we did not have to make a function. Instead we wrote the option in the main function. First we opened the writing file using the same name as the read file, then we used for loop on the list to write every student info, and then we used if statement to check if the course not empty so we can avoid writing comma after student info, then we rite the comma between student info and courses, then we used another two for loop inside each other, first to get every course and the second to get info per course. Then we write the course info, and then we write comma after info only if it’s not the last course for the student. And then we write semi-column after each course only if it’s not the last course. And then we used if statement to write a new line for another student only if it’s not the last student. And finally, we close the file to save changes.



# NOTE: every time we write check that means we show appropriate message if the entry is wrong or not found.