**Assignment 1**

**Special Instructions:**

* **All assignment shall be submitted by the deadline.** Late submissions will be penalized with 10% per day for up to 3 calendar days after which the assignment cannot be submitted anymore. An **email must be sent** should you choose to submit a late assignment.
* This assignment is an individual work.
* Even partially copied code will be subject to regulations against academic integrity. Do NOT discuss or share your solution with anybody. Posting this assignment or solution on the Internet is a violation of the Student Code of Conduct. See the Academic Honesty at Sheridan.
* Deductions will be applied if partial functionality is provided.
* You’ll get zero grade if your code doesn’t compile.
* Compilation warnings are considered to be major mistakes.
* Copy your C program to text file named [*PROG20799\_ASG1<Student number>\_<lastName>.txt*] containing your code and include a program header (i.e., a block comment at the top with your name, date, description of program).
* A well-documented program. Make sure your program has no compilation error and is correct.
* Submission is done in electronic format **using SLATE DropBox. DO NOT email your submission.**
* This assignment has 1 question1 on 4 pages. The assignment is out of 100.

*In this assignment, you are required to implement different functions to handle a List Data Structure implemented using array in C.*

You are required to use array of structures and files in C by developing a GPA Calculation Application. The application should maintain and store course grades as a list of courses for a student. The application should display a menu of operations that handles update, find maximum grade, display, compute the accumulative GPA[check appendix about how to calculate GPA ], and save the grades to the file course grades.

1. Create a structure data type called **Course** that contains the following information: Course ID, Course Credit and Grade. You need to implement the following:
2. courseID as a sequence of characters (e.g. Prog20799).
3. credits as an integer that represents the credit of the course (e.g. 6).
4. grade as a float that represents the grade of the course(e.g. 85)
5. Create an array of 4 courses. Please use define MAX\_COURSES 4.
6. Read 4 courses from the file(grades.txt) in the main function before calling the menu function. The input file called grades.txt is attached with the assignment.
7. Perform Menu-Operations:
8. Update a course grade in the list of course.
   1. void updateCourseGrade(Course courses[])
9. Find Maximum Grade. This is a recursive function and you need to implement
   1. int max(int n1, int n2)
   2. float findMaxGrade(Course courses[], int n)
10. Display a list of courses taken by the student.
    1. void display(Course courses[])
11. Compute the accumulative GPA for all courses.
    1. float gradePoints(float grade) - return the grade point of a grade e.g. if grade = 90 then grade point = 4.
    2. float computeGPA(Course courses[])- return GPA for all courses
12. Save all courses and grades in an output file called “output.txt”
    1. void saveCourses(char str[], Course courses[])

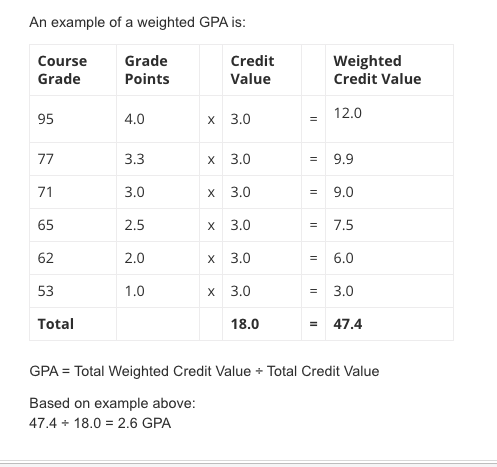
**Evaluation**

|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 Grading Page | | | |
|  | | Max Mark | Mark |
| Code[100 marks] | Documentation including a Program header with names and simple comments for all functions | 10 |  |
| Implementation of Data Types | 10 |  |
| Perform read from the file -(grades.txt) | 10 |  |
| Perform Update Grade and validation | 15 |  |
| Perform print the courses | 10 |  |
| Perform save the courses-(output.txt) | 10 |  |
| Perform Max and Recursive find Max grade | 15 |  |
| Perform compute GPA | 20 |  |
|  | Fail to follow submission guidelines(-10%) | 0 |  |
|  | No warning during compilation (-10%) | 0 |  |
|  | Late submission -10% per day | 0 |  |
| Total |  | 100 |  |
| Comments |  | | |

**Appendix:**

**GPA Calculation [[1]](#footnote-1)**





1. <http://myotr.sheridancollege.ca/grading_systems.html>

   <http://myotr.sheridancollege.ca/gpa.html> [↑](#footnote-ref-1)