

Assignment 01

Index: 23CDS0843

Problem:

Write an algorithm in pseudocode to calculate a student's Final Grade Point Average (FGPA) based on the grades and credit values of all registered GPA courses, in accordance with the GPA calculation rules outlined in Pages 90–91 of the Student Handbook 2023/2024 (Faculty of Computing).

Pre-notes:

In the handbook it defines **N** as the **total number of credits assigned per year**. so this is simply means

$$N = \sum_{i=1}^n CP_i$$

$$GPA = \frac{\sum CP_i \times GP_i}{\sum CP_i}$$

$$FGPA = \sum_{j=1}^4 (a_j \times P_j)$$

Definitions & Assumptions:

- Each **COURSE** will have two properties:
 - *credits* (numerical value),
 - *attempts* (list of numerical marks of each attempt - where the last one is the most recent)
- **LAST** - Assumed function that runs on any list that returns the last element of that given list.
- **LENGTH** - A function that returns length of an object (like length of a list)
- **ROUND** - A function that rounds a value to a given decimals points
- In code `sum_CPxGP` refers to the part $\sum CP_i \times GP_i$
- In code `total_cp` refers to that N or $\sum CP_i$
- Known information is that `total_cp` is always **Non-Zero** (`total_cp > 0`) since N never be 0. (mentioned to avoid dividing by zero)
- Handbook rounded the FGPA to 2 decimals

ALGORITHM:

INPUT:

```
For each year y in (1,2,3,4):  
    Courses[y] = list of GPA courses taken in year y
```

OUTPUT: FGPA value, rounded to 2 decimal points

DEFINED CONSTANTS:

```
year_weight[1] = 0.2  
year_weight[2] = 0.2  
year_weight[3] = 0.3  
year_weight[4] = 0.3
```

```
A_plus = 4.00  
A = 4.00  
A_minus = 3.70  
B_plus = 3.30  
B = 3.00  
B_minus = 2.70  
C_plus = 2.30  
C = 2.00  
C_minus = 1.70  
D_plus = 1.30  
D = 1.00  
E = 0.00
```

// Function that returns the grade point from marks (using the handbook table)

FUNCTION GRADE_FROM_MARKS(Mark m):

```
IF m >= 90 RETURN A_plus  
ELSE IF 80 <= m <= 89 RETURN A  
ELSE IF 75 <= m <= 79 RETURN A_minus  
ELSE IF 70 <= m <= 74 RETURN B_plus  
ELSE IF 65 <= m <= 69 RETURN B  
ELSE IF 60 <= m <= 64 RETURN B_minus  
ELSE IF 55 <= m <= 59 RETURN C_plus  
ELSE IF 50 <= m <= 54 RETURN C  
ELSE IF 45 <= m <= 49 RETURN C_minus  
ELSE IF 40 <= m <= 44 RETURN D_plus  
ELSE IF 30 <= m <= 39 RETURN D  
ELSE RETURN E
```

```

// Function to account the repated cases
FUNCTION PROPER_GRADE(Course c):
    last_attempt = LAST(c.attempts)

    last_grade = GRADE_FROM_MARKS(last_attempt)

    // regardless of grade/marks, repeated exams get C if last attempt is (D or
    above, in other words if didn't get an E)

    IF LENGTH(c.attempts) >= 2 AND last_grade != E THEN
        RETURN C
    ELSE
        RETURN last_grade

// Starts of the calculations with the help of above helper functions.

MAIN_FUNCTION:

FGPA = 0
FOR y = 1 TO 4 DO
    sum_CPxGP[y] = 0
    total_cp[y] = 0

    FOR EACH Course c IN Courses[y] DO
        gp_value = PROPER_GRADE(c)

        sum_CPxGP[y] = sum_CPxGP[y] + (c.credits * gp_value)
        total_cp[y] = total_cp[y] + c.credits
    END FOR

    year_GPA[y] = sum_CPxGP[y] / total_cp[y]
    FGPA = FGPA + (year_weight[y] * year_GPA[y])
END FOR
RETURN ROUND(FGPA, 2)

```

I wrote & compiled this document with Typst (a LaTeX alternative): typst.app