Benisha Devalla, Emily Zhang, Nikki Somasundram, Rachel Yin

T3 CSA PROJECT

Rachel - Rachel Yin
Nikki - ROSHINIKITHA SOMASUNDRAM
Emily - EMILY ZHANG
Benisha - Benisha Devalla

FINAL PSEUDOCODE SUBMISSION

(Taking attendance, scheduling a student for courses, and assigning and reporting grades, with file I/O, with AP / no AP courses, more error handling, more looping, adding and calculating more grades) (+ signs are methods / behaviors) (- signs are variables / attributes)

Class: Main

```
-   StudentRosterID: array of 5 ints - unique ID only
-   StudentRoster: Array of 5 strings - names that correspond to the Unique ID
        -   (StudentRosterID and StudentRoster are parallel arrays)
-   Int UniqueID = 000;
+   Menu() - loop until exit option selected
        +   prompt user (what would you like to do) (register student, add student to class,
            mark attendance, update grade)
        +   If register student
                +   PrintStudents()
                +   Call registerStudent()
                +   Loop until exit option is selected
        +   If add student to course
                +   PrintStudents()
                +   Call AddStudentToCourse()
                +    Loop until exit option is selected
        +   If update attendance
                +   ShowAttendance()
                +   ask if you want to update attendance
                +   If yes
                        +   Ask user to Choose a student - based on ID
                        +   UpdateAttendance() / call the update attendance and store the
                            updated in attendance + add a exit option
                        +   Return to all student name menu
                        +   Loop to 'If update attendance' until exit option selected
                +   If no
                        +   Exit
        +   If update grade
                +   PrintStudents()
                +   Ask user to Choose a student - based on ID
                +   StudentStats()
                +   Call Grade() + add a exit option
                +   Return to all student name menu
                +   Loop until exit option selected
```

Benisha Devalla, Emily Zhang, Nikki Somasundram, Rachel Yin

```
        +   Error handle
            +   If user asks for a student that doesn't exist, or tries to register

    +   registerStudent()
        +   Asks user for student (First name, last name, graduation year, attendance
            counts, course, grade stats)
        +   StudentUniqueID = ++UniqueID; - make the student's unique ID, the current ID,
            then add to it so the next student doesn't have the same ID
        +   Within this step call AddStudentToCourse()
        +   Makes a new file with student data() and sets file name as student's Unique ID
        +   puts the UniqueID in an array for later reference
    +   PrintStudents()
        +   Print the 1st Name and 1st ID of the two arrays
        +   Loop until both arrays are at end
        +   Exit.
    +   AddStudentToCourse()
        +   PrintStudents() and prompt user to pick one student - based on ID
        +   If student doesn't exist
            +   Throw error and call AddStudentToCourse()
        +   If student does exist, proceed
        +   StudentStats() - prints it
        +   If student is already in course
            +   Throw error and ask user to try again
        +   If student is not in course - ask what course
            +   Add the course to student file ( World History, Geormetry, AP chem, AP
                LIst)
                +   Example: World History
                    +   Course name: World
                    +   Course Type: int 0 - no AP
                    +   Course overall grade: char
                        +   quizzes percentage score: int
                            +   Quiz 1,2,3,4,5 scores (20 points): int
                        +   One midterm percentage score: int
                            +   Midterm score (50 points): int
                        +   One final percentage score: int
                            +   Final score  (100 points): int
                        +   Overall percentage grade: int
                    +   Course Attendance
                        +   (1) Present: int
                        +   (2) tardy: int
                        +   (3) absent: int
                +   Repeat for Geometry, AP Chem, AP Lit


*after AddStudentToCourse(course1)*
    -   Course name: String
    -   Course overall grade: char
        -   quizzes percentage score: int
            -   Quiz 1,2,3,4,5 scores (20 points): int
            -   One midterm percentage score: int
```

        - Midterm score (50 points): int
    - One final percentage score: int
    - Final score (100 points): int
    - Overall percentage grade: int
- Course Attendance
    - (1) Present: int
    - (2) tardy: int
    - (3) absent: int

<u>Class: Student</u>

- First Name: string
- Last Name: String
- Graduation year: int
- Unique ID: Int
- Overall Attendance (sum of all the presents, tardies, and absences a student has)
    - (1) Present: int
    - (2) tardy: int
    - (3) absent: int
- Course Enrolled: array of strings or file list

+ StudentStats(int FileDisplayType)
    + Reads student file
    + Prompts if user want full or short file
    + If FileDisplayType == 0
        + Prints out (first name, last name, ID, attendance counts, course name and letter grade only)
    + If FileDisplayType == 1
        + Prints out (First name, last name, ID, Grad year, attendance counts, course details (name, assignments, attendance, ect))

+ UpdateAttendance()
    + StudentStats(0) or (print out student name, and attendance counts only)
    + Prompts user for what course to count it for
    + Prompts user to select what type
        + print out a display that talks about entering attendance
        + Print “1 for present, 2 for tardy, and 3 for absent”
        + If a valid number (1,2,3) (add to: present, tardy, absent)
            + if option 1 is selected then student attendance is present (add 1 to present count for that course)
            + Else if option 2 is selected then student attendance is tardy (add 1 to present count for that course)
            + else if option 3 is selected then student attendance is absent (add 1 to present count for that course)
        + If any other value, throw error and try again
    + Recalculate total attendance by taking course attendance and adding them up
    + Updates the file (adds 1 to selected count), StudentStats(0), and exits

Benisha Devalla, Emily Zhang, Nikki Somasundram, Rachel Yin

```
+ ShowAttendance()
    + Loop for all students on the file / StudentRoster array
        + Print name
        + Print overall Attendance / print / show updated attendance of student
            + (1) Present: int
            + (2) tardy: int
            + (3) absent: int
    + Exit

+ Grade()
    + StudentStats(1) or (prints out student name and grade stats)
    + Prompts user for what type of input they want to do (Override, normal grade)
    + If Override
        + Prompt user for what to override (Quiz, Homework, Midterm, final, overall
          grade)
        + Prompt user to only override percentages
        + Prompt user what they would like to override it to
            + Percentage - int only input , 0 <= input <= 100
    + If normal grade
        + Prompt user for what to grade (Quiz, Homework, Midterm, final)
        + If quiz
            +  Prompt user for what quiz to change
            + Ask user what score they would like to assign
                + Score must be int, 0 <= input <= 20
        + If homework
            + Ask if all homework thus far has been submitted
            + If yes, make homework = 100 (%)
            + If no, make homework = 0 (%)
        + If midterm
            + Ask user what score they would like to assign
                + Score must be int, 0 <= input <= 50
        + If final
            + Ask user what score they would like to assign
                + Score must be int, 0 <= input <= 100
    + Updates file, CalculateGrade(), StudentStats(1)
+ CalculateGrade()
- run each time file is updated /StudentsStats() is called in other functions
    + StudentStats(2) or (prints out student name and grade stats)
    + Read in course assignment scores
    + If - some or all assignments are graded, some or none are blank - calculates
      score and sets it to Course overall grade
        + Loop for each assignment score
            + (if the score is blank, do not take any values,
            + if the score is full, take the value of the percentage and add that in
              the calculations)
        + Reads in course Type
            + If AP
                + Set weights to:
                    + Quiz: 10%
                    + Summative exams: 80%
                    + Homework: 10%
            + If non-AP
                + Set weights to:
                    + Quiz: 20%
                    + Summative exams: 80%
                    + Homework: 0%
        + Selects only assignments that are not blank
        + Calculates each percentage, overall, and letter score.
        + Updates the file based on those stats
        + Loop until all courses are done
    + If - no assignments are graded, all are blank - set final grade to N/A
        + Return N/A - or no grade available
    + StudentStats(1) - reprints out
```