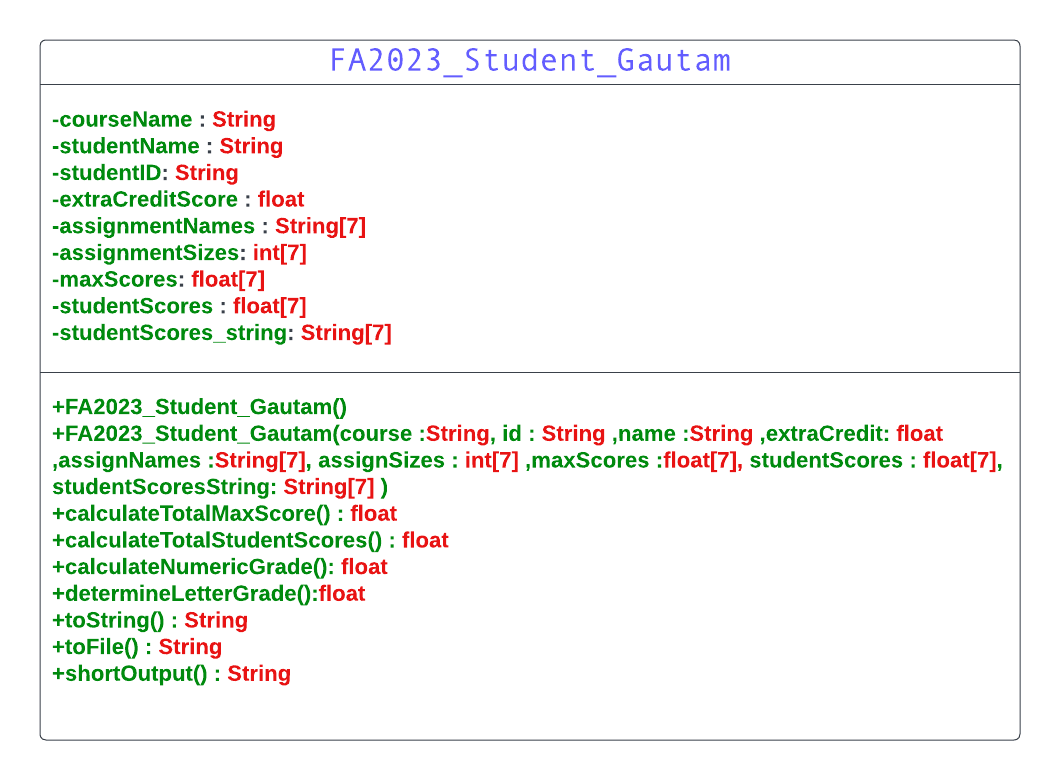
**<<<<<<<<UML DIAGRAM OF DATA TYPE CLASS>>>>>>>>>>><<<<<<<<<<PSEUDO CODE FOR DATA TYPE CLASS>>>>>>>>>>>>>**

**FA2023\_Student\_Gautam.java**

class FA2023\_Student\_Gautam {

// Data Members

private String courseName

private String studentName

private String studentID

private float extraCreditScore

private String[] assignmentNames

private int[] assignmentSizes

private float[] maxScores

private float[] studentScores

private String[] studentScoresString

// Constructors

FA2023\_Student\_Gautam() {

// Default constructor

}

FA2023\_Student\_Gautam(course, id, name, extraCredit, assignNames, assignSizes, maxScores, studentScores, studentScoresString) {

// Parameterized constructor

}

// Methods

// Calculate total max score

function calculateTotalMaxScore(): float {

totalMaxScore = 0

for each element in maxScores:

totalMaxScore += element

return totalMaxScore

}

// Calculate total student scores

function calculateTotalStudentScores(): float {

totalStudentScores = 0

for each element in studentScores:

totalStudentScores += element

return totalStudentScores + extraCreditScore

}

// Calculate numeric grade

function calculateNumericGrade(): float {

totalMaxScore = calculateTotalMaxScore()

totalStudentScores = calculateTotalStudentScores()

numericGrade = (totalStudentScores / totalMaxScore) \* 100

return numericGrade

}

// Determine letter grade

function determineLetterGrade(): String {

numericGrade = calculateNumericGrade()

if (numericGrade >= 90) {

return "A"

} else if (numericGrade >= 80) {

return "B"

} else if (numericGrade >= 70) {

return "C"

} else if (numericGrade >= 60) {

return "D"

} else {

return "F"

}

}

// Convert scores to string

function scoresToString(scores: float[]): String {

scoresString = ""

for each score in scores:

scoresString += score + " "

return scoresString

}

// Display information in the specified format

function toString(): String {

result = "FA2023\_FinalGradeApplication\_Gautam\n"

result += "FINAL GRADE OF STUDENT - DIPESH GAUTAM\n"

result += "---------------------------------------------\n"

result += "COURSE NAME: " + courseName + "\n"

result += "STUDENT ID: " + studentID + "\n"

result += "STUDENT NAME: " + studentName + "\n"

result += "---------------------------------------------\n"

result += "EXTRA CREDIT: " + extraCreditScore + "\n"

// Display scores for each assignment type

for i from 0 to assignmentNames.length - 1:

result += assignmentNames[i] + "\t\t" + scoresToString(studentScores[i]) + "\n"

result += "---------------------------------------------\n"

result += "TOTAL SCORES: " + calculateTotalStudentScores() + "/" + calculateTotalMaxScore() + "\n"

result += "NUMERIC GRADE: " + calculateNumericGrade() + "\n"

result += "LETTER GRADE: " + determineLetterGrade() + "\n"

return result

}

// Convert information to a string for file output

function toFile(): String {

result = courseName + "," + studentID + "," + studentName + ","

result += calculateNumericGrade() + "," + determineLetterGrade() + ","

result += extraCreditScore + ","

// Append scores for each assignment type

for i from 0 to assignmentNames.length - 1:

result += scoresToString(studentScores[i]) + ","

return result

}

// Display a shortened version of the output

function shortOutput(): String {

result = "STUDENT: " + studentID + "\n"

result += "NAME: " + studentName + "\n"

result += "Numeric Grade: " + calculateNumericGrade() + "\n"

result += "Letter Grade: " + determineLetterGrade() + "\n"

return result

}

}

**<<<<<<<<<<PSEUDO CODE FOR DRIVER CLASS>>>>>>>>>>>>>>>>**

**FA2023\_FinalGradeApplication\_Gautam.java**

// Constant Arrays

const int SIZE = 7;

const String ASSIGNMENT\_NAMES[SIZE] = {"QUIZ", "HOMEWORK", "LAB", "PROJECT", "TEAMWORK", "DISCUSSION TOPIC", "TEST"};

int assignmentSizes[SIZE];

float maxScores[SIZE];

// Function to Read Assignment Sizes

function readAssignmentSizes():

for i in 0 to SIZE - 1:

print "Enter the number of assignments for", ASSIGNMENT\_NAMES[i], ": "

assignmentSizes[i] = readInt()

// Function to Read Max Scores

function readMaxScores():

for i in 0 to SIZE - 1:

print "Enter the maximum score for", ASSIGNMENT\_NAMES[i], ": "

maxScores[i] = readFloat()

// Function to Display Menu

function displayMenu():

print "FA2023\_FinalGradeApplication\_Gautam"

print "TASK OF GRADING - DIPESH GAUTAM"

print "------------------------------------"

print "1. Grading One Student"

print "2. Printing The Grade of One Student from Input File"

print "3. Printing The Grades of Class"

print "0. Exit"

// Function for Task 1: Grading One Student

function gradingOneStudent():

// Read information and scores from the keyboard

// Create an object of FA2023\_Student\_Gautam

// Access the toString() method to display the grade

// Write to file using the toFile() method

// Function for Task 2: Printing The Grade of One Student from Input File

function printGradeFromInputFile():

// Read student ID from the keyboard

// Read from the input file "StudentGrade.txt"

// For each line, compare student ID and display the grade if there is a match

// Function for Task 3: Printing The Grades of Class

function printGradesOfClass():

// Read grades from the file "StudentGrade.txt"

// For each line, create an object of FA2023\_Student\_Gautam

// Access the method to display the grades in the requested format

// Main Program

readAssignmentSizes()

readMaxScores()

// Loop for Menu

do:

displayMenu()

choice = readInt()

switch choice:

case 1:

gradingOneStudent()

break

case 2:

printGradeFromInputFile()

break

case 3:

printGradesOfClass()

break

case 0:

print "Exiting the program."

break

default:

print "Invalid choice. Please try again."

while choice != 0