Credit Name: CSE 2120 Data Structures 1
Assignment Name: Chapter 9 CourseGrades

How has your program changed from planning to coding to now? Please explain?

In my gradeBook class, I began my importing scanner to prepare for user input

```
//Preparing for user input
static Scanner input = new Scanner(System.in);
```

Initalized final int variables students and tests; to remain consistant throughout runtime.

```
//Initialize final variables
static final int students = 12;
static final int tests = 5;
```

Initalized and created a 2-d array to store test grades in accordance to each student. Array contained #student rows, and #tests columes within.

```
//Initialize and create 2-d array to store grades;
//with 12 rows (students), and 5 columns (tests)
private static int[][] grades = new int[students][tests];
```

Created a method that retrieves each test grade in relation to each sudent. Method contains no parameters, and does not return a value.

```
//This method gets the grades for each student from the teacher
//@param n/a
//@return: void
public static void getGrades(){
```

In the method, for each student (element) within the variable student, (starting at the first element),

```
//for each student in the students variable
for (int s = 0; s < students; s ++) {
```

Display a section header to user to indicate which student's grades to follow:

```
//section header
System.out.println(" ");
System.out.println(" ");
System.out.println("Please enter the following test grades(%) for student #" + (s + 1) + " :");
System.out.println(" ");
```

For each test (element) within the test variable, (starting at first element),

```
//for each test in the tests variable
for (int g = 0; g < tests; g++ ) {
```

Prompt the user for the correspondent grade for the test #, and record user's input as the correspondent index in the 2-D array.

```
//Prompt user for grades for each test + record user input into array
System.out.print("Test " + (g + 1) + ": ");
grades[s][g] = input.nextInt();
```

Created a method that displays each test grade in relation to each sudent. Method contains no parameters, and does not return a value.

```
//This method gets the grades recorded, and outputs them to user
    //@param n/a
    //@return: void
public static void showGrades() {
```

Display section header to inform user of following information output.

```
//section header
System.out.println(" ");
System.out.println("Class Grades:");
```

In the method, for each student (element) within the variable student, (starting at the first element),

```
//for each student in the students variable,
for (int s = 0; s < students; s ++) {
```

Display a section header to user to indicate which student's grades to follow:

```
//section header
System.out.println(" ");
System.out.println(" ");
System.out.println("Student # " + (s + 1) + " :");
System.out.println(" ");
```

For each test (element) within the test variable, (starting at first element),

```
//for each test in the tests variable,
for (int g = 0; g < tests; g++ ) {
```

Display corresponding grades in index of array (entered by user) back to screen

```
//output test grades to user for correspondent grade
System.out.println("Test " + (g + 1) + ": " + (grades[s][g]) + "%");
```

Created a method that calculates the average test grade of each student, and returns value.

Method contains int parameters (student num - specified student), and returns a double value (avg test grade of student)

```
//This method calculates the average test score of each user and returns the value
    //@param (int) student number: each specific student
    //@return(double): sum of all grades of student, divided by number of tests
public static double studentAvg(int studentNumber) {
```

Within method, initalize double variable avg to store sum of test grades.

```
//initialize variable to store tests sum
double avg = 0;
```

For each test (element) within the test variable, (starting at first element),

```
//for each test in the tests variable,
for (int g = 0; g < tests; g ++) {
```

Set variable avg to a new value of the original sum plus the value of the array element at the correspondent index of test g, and the student num param.

```
//Set sum equal to the current value of sum, plus the test grade at of test g at studentNumber
avg = avg + (grades[studentNumber][g]);
```

Divide the value of avg by the number of tests, and return the value.

```
 //Divide sum of tests by # of tests; return value.
 return (avg/tests);
```

Created a method that calculates the average grade of a test among all students and returns value.

Method contains int parameters (testnum - specified test), and returns a double value (avg test grade of a test among all students)

```
//This method calculates the average grade of a test among all students.
    //@param (int) test number: each specific test
    //@return(double): sum of all grades of a test, divided by number of students
public static double testAvg(int testNumber) {
```

Within method, initalize double variable avg to store sum of test grades.

```
//initialize variable to store tests sum
double avg = 0;
```

In the method, for each student (element) within the variable student, (starting at the first element),

```
//for each student in the students variable,
for (int s = 0; s < students; s ++) {
```

Set variable avg to a new value of the original sum plus the value of the array element at the correspondent index of student s, and the test num param.

```
//Set sum equal to the current value of sum, plus the test grade at index testNumber
avg = avg + (grades[s][testNumber]);
```

Divide the value of avg by the number of students, and return the value.

```
//Divide sum of tests by # of students; return value.
return (avg/students);
```

In the main method, I began by creating decimal formating to shorten longer decimals to 2 decimal places,

```
public static void main(String[] args) {

    //format decimal variables
    DecimalFormat shorten = new DecimalFormat("#00.00");
```

Using the getGrades method, I retrieved the grades from the user.

Using the showGrades method, I displayed the previously entered grades back to the user.

```
//Get grades from user using method
getGrades();

//Display grades to user using method
showGrades();
```

Display a section header to user to indicate which student grade avgs are outputted.

```
//Section header
System.out.println(" ");
System.out.println("Average test grade for each student:");
System.out.println(" ");
```

For each student (element) within the variable student, (starting at the first element),

Display the student #, and their correspondent avg, using the studentAvg method to retrieve result. Format result using decimal format to 2 decimal places.

```
//for each student in the students variable,
for (int i = 0; i<students; i++) {

    //Output student number, and student Avg using method; decimal format result.
    System.out.println("Student " + (i+1) + ": " + shorten.format(studentAvg(i)) + "%");

}
```

Display a section header to user to indicate which test grade avgs are outputted.

```
//Section header
System.out.println(" ");
System.out.println("Class average for each test:");
System.out.println(" ");
```

For each test (element) within the variable test, (starting at the first element),

Display the test #, and the correspondent avg, using the testAvg method to retrieve result. Format result using decimal format to 2 decimal places.

```
//for each test in the tests variable,
for (int i = 0; i<tests; i++) {

    //Output test number, and test Avg using method; decimal format result.
    System.out.println("Test " + (i+1) + ": " + shorten.format(testAvg(i)) + "%" );
```