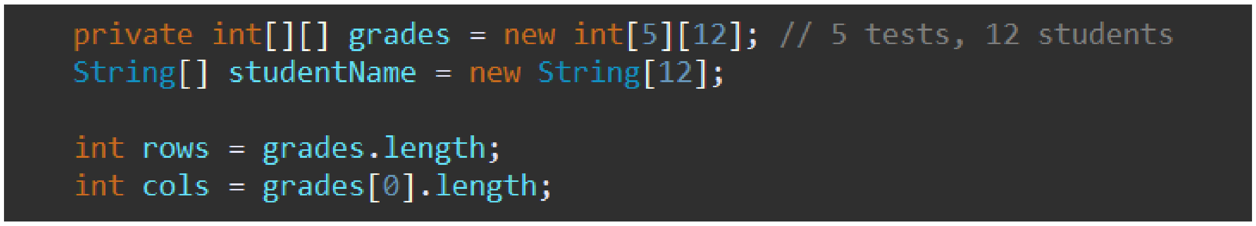
Credit Name: CSE 2120 Data Structures 1

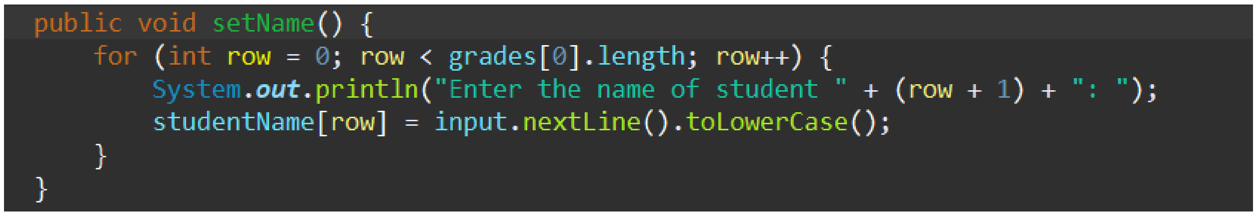
Assignment: Course Grades

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



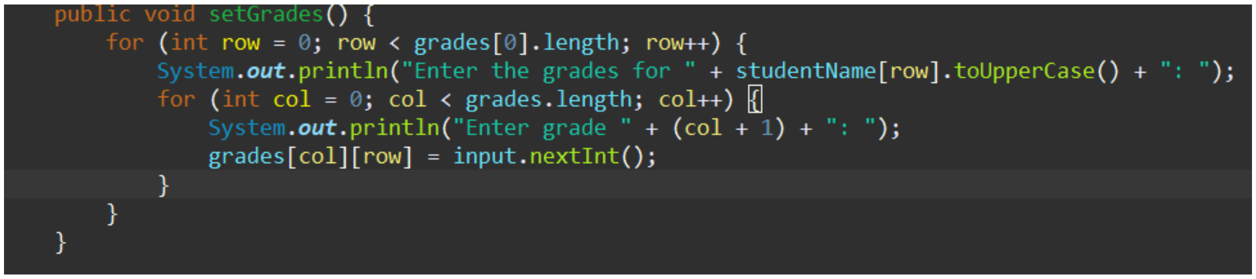
```java
    private int[][] grades = new int[5][12]; // 5 tests, 12 students
    String[] studentName = new String[12];

    int rows = grades.length;
    int cols = grades[0].length;
```

Firstly, I coded my gradebook object.I started by making a 2D array for the grades. There are 5 rows for the 5 different tests and 12 columns for the 12 students in the class. I then made a list for all the students' names. I then defined the number of rows and columns using the .length method.



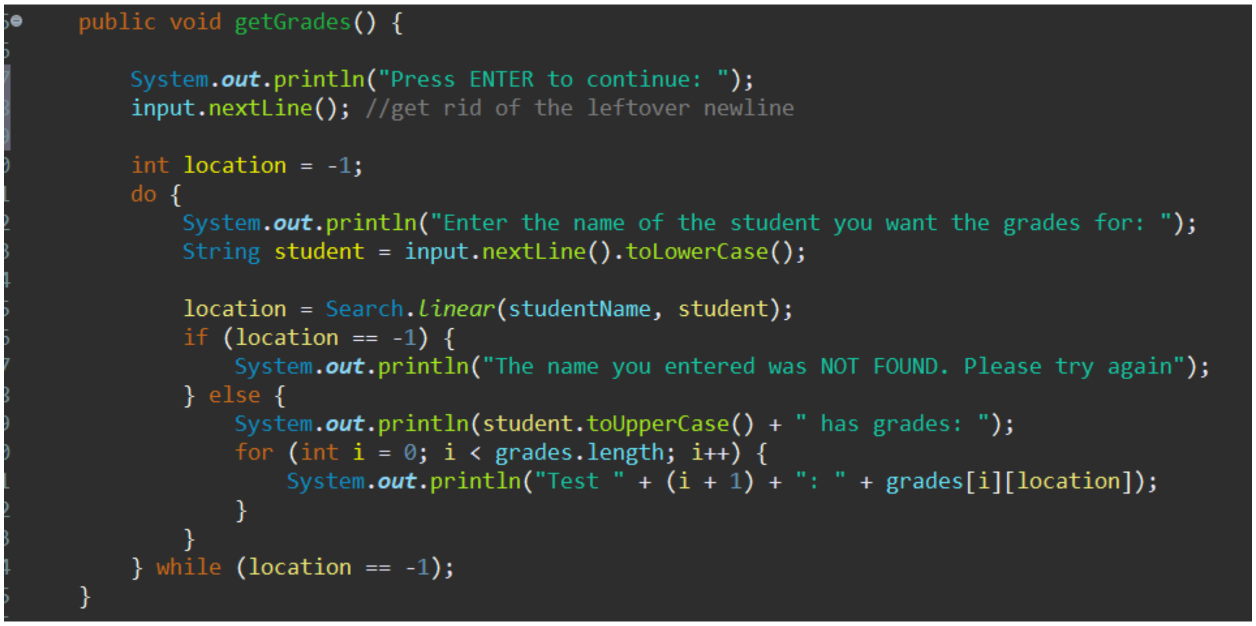
```java
  public void setName() {
      for (int row = 0; row < grades[0].length; row++) {
          System.out.println("Enter the name of student " + (row + 1) + ": ");
          studentName[row] = input.nextLine().toLowerCase();
      }
  }
```

The first method i made was to set the names of the students. I made a for loop and set it equal to the number of columns/students and asked the user to input the name of the student and indicated the order in which they are typed into the system to make it easier to keep track of the students added in. The names are saved in lower case so that it is not type sensitive. Like this the studentName list is filled with the name of the various students.



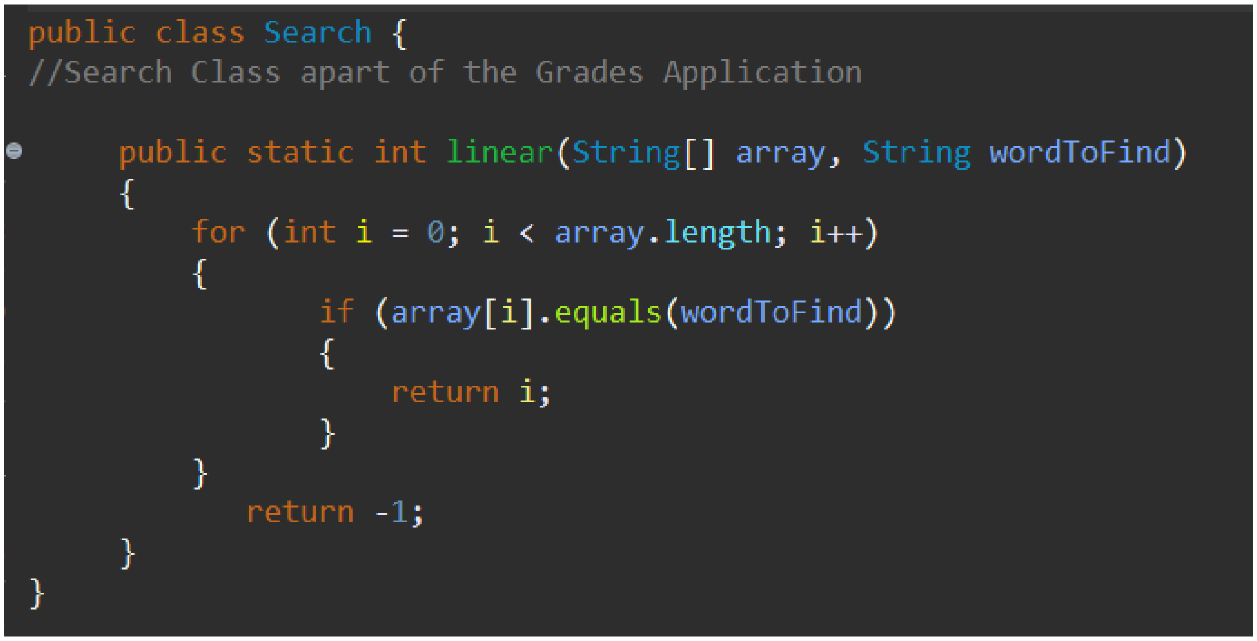
```java
public void setGrades() {
    for (int row = 0; row < grades[0].length; row++) {
        System.out.println("Enter the grades for " + studentName[row].toUpperCase() + ": ");
        for (int col = 0; col < grades.length; col++) {
            System.out.println("Enter grade " + (col + 1) + ": ");
            grades[col][row] = input.nextInt();
        }
    }
}
```

The next method is to to set the grades of each student for each of the five tests. I started with a for loop which goes through the 12 students' names and asks for their test score. Then within that for loop there is another for loop which continues for 5 iterations, 1 for each test so for each student there are 5 test scores inputted. As the scores are inputted they are saved in the appropriate row and col of the grades array.



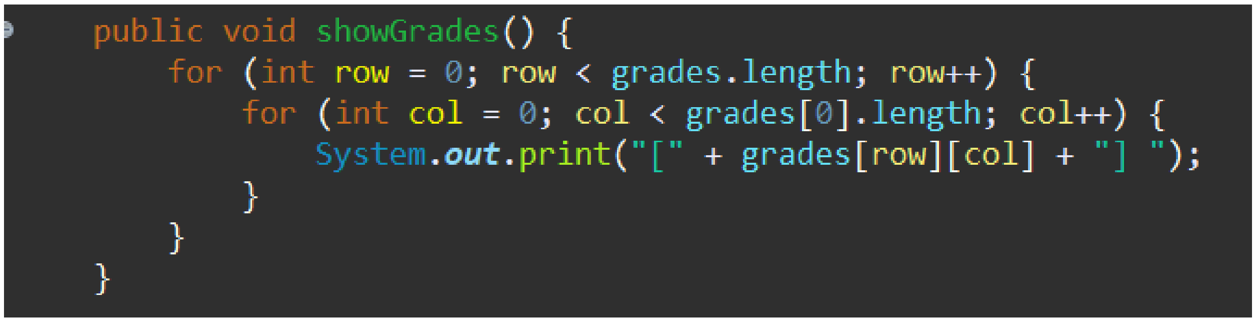
```java
public void getGrades() {

    System.out.println("Press ENTER to continue: ");
    input.nextLine(); //get rid of the leftover newline

    int location = -1;
    do {
        System.out.println("Enter the name of the student you want the grades for: ");
        String student = input.nextLine().toLowerCase();

        location = Search.linear(studentName, student);
        if (location == -1) {
            System.out.println("The name you entered was NOT FOUND. Please try again");
        } else {
            System.out.println(student.toUpperCase() + " has grades: ");
            for (int i = 0; i < grades.length; i++) {
                System.out.println("Test " + (i + 1) + ": " + grades[i][location]);
            }
        }
    } while (location == -1);
}
```

The third method is one which gets the grades of the one students for all 5 tests. Firstly I needed to add an input at the beginning because there is leftover newline which disrupts the remainder of the method. Then I get the user to input the name of the student they are looking for. It converts it to lower case to match the studentName list. Then using the search class i coded the location(index) of the students name is searched for from the studentName list. If the name was not found by the search function it returns a value of -1 which is then indicated in a print statement. Otherwise a print statement highlights the student's whos grades are about to be shown. And then the grades are printed for the student by using the constant location value for the student and using a for loop for the 5 test scores. The loop will keep repeating while until a valid name is entered.



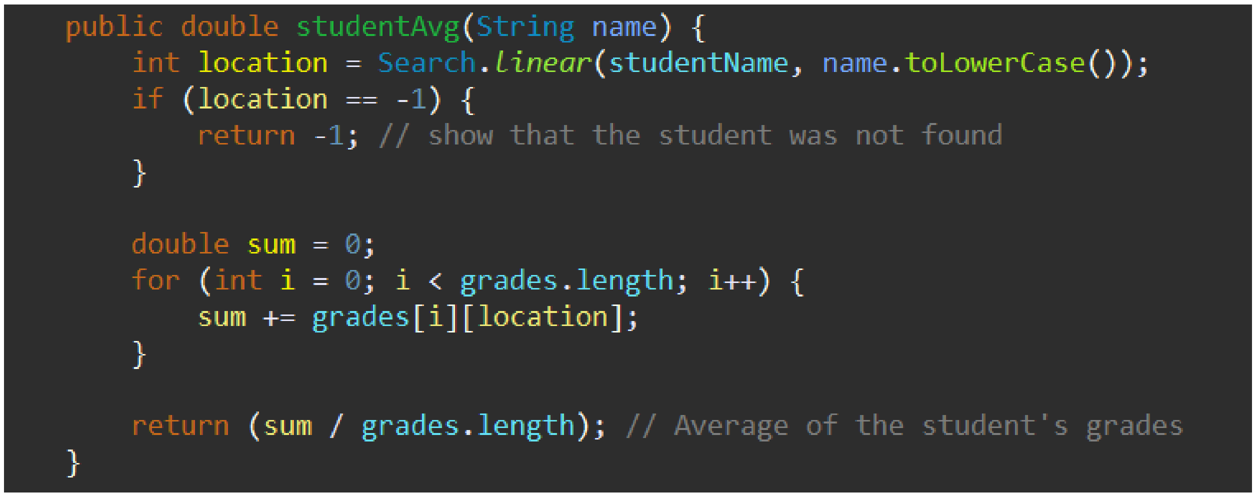
```java
public class Search {
//Search Class apart of the Grades Application

        public static int linear(String[] array, String wordToFind)
        {
            for (int i = 0; i < array.length; i++)
            {
                    if (array[i].equals(wordToFind))
                    {
                        return i;
                    }
            }
                return -1;
        }
}
```

The Search class has a linear method which takes in the given list and the word you want to find in the list. A for loop and a if statement is used to see if any element in the array is the same as the word you need to find. If the word is found then the index that it was found at was returned. Otherwise -1 is returned to show it wasn't found.



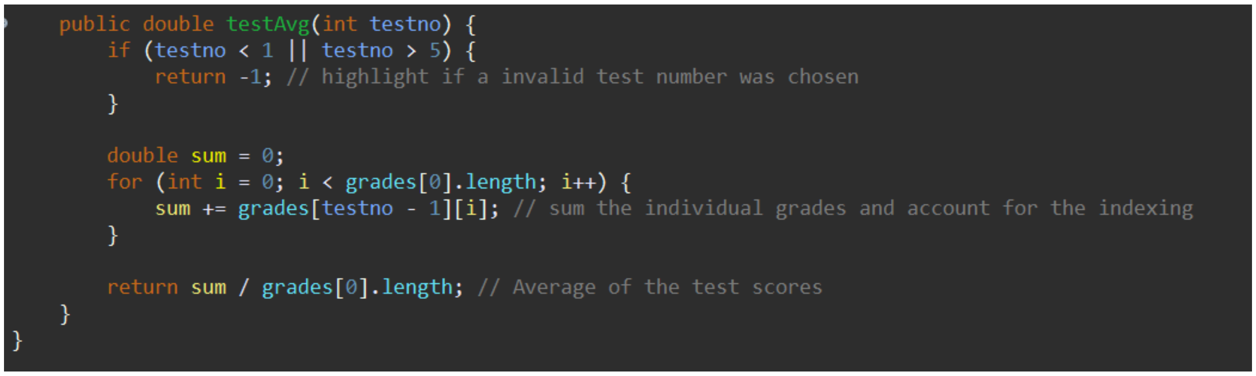
```java
    public void showGrades() {
        for (int row = 0; row < grades.length; row++) {
            for (int col = 0; col < grades[0].length; col++) {
                System.out.print("[" + grades[row][col] + "] ");
            }
        }
    }
```

The next method is a simple spreadsheet of the scores the students achieved for each row (test) every single grade for each column(student) is printed between square brackets.



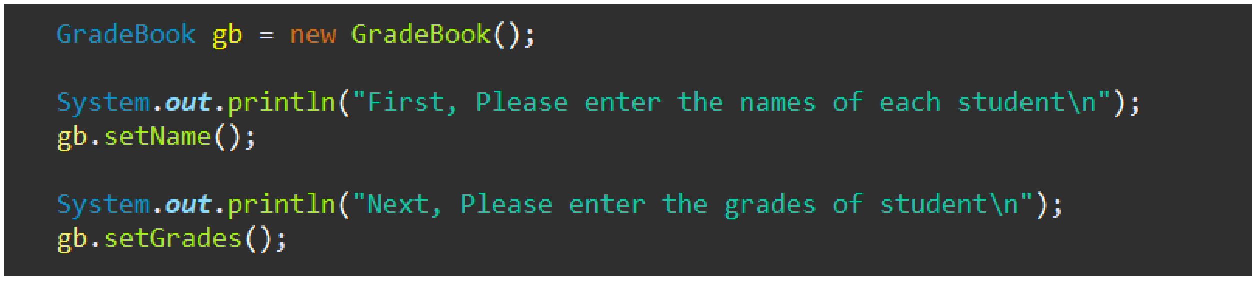
```java
    public double studentAvg(String name) {
        int location = Search.linear(studentName, name.toLowerCase());
        if (location == -1) {
            return -1; // show that the student was not found
        }

        double sum = 0;
        for (int i = 0; i < grades.length; i++) {
            sum += grades[i][location];
        }

        return (sum / grades.length); // Average of the student's grades
    }
```

The penultimate method initially uses the search class and linear method to identify the name that was given in the studentAvg's parameter. If location was not found(= -1) then the student avg is also given as -1 which will indicate in the client code that the name was not found. Otherwise the sum is calculated by using a for loop to find each test grade for that specific student and to keep adding it to the sum which was initialized to 0. Then the sum/ no. of tests is returned which is the average grade for the student.



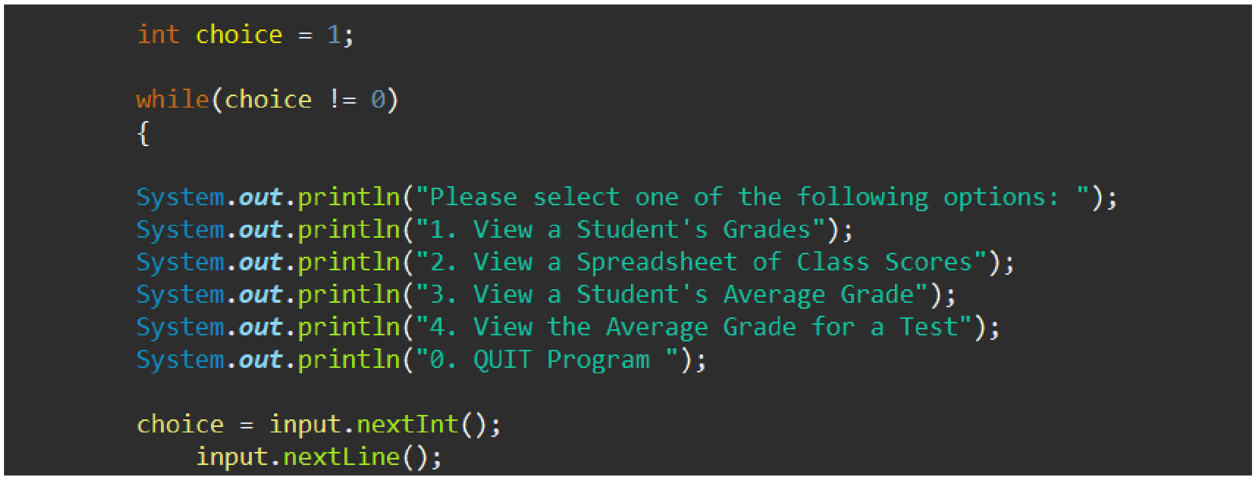
```java
    public double testAvg(int testno) {
        if (testno < 1 || testno > 5) {
            return -1; // highlight if a invalid test number was chosen
        }

        double sum = 0;
        for (int i = 0; i < grades[0].length; i++) {
            sum += grades[testno - 1][i]; // sum the individual grades and account for the indexing
        }

        return sum / grades[0].length; // Average of the test scores
    }
}
```

The last method has an int parameter which corresponds to which test they want to look at (test 1-test 5)
If the number they input isn't between 1 and 5, then again -1 is returned which indicates invalid input.
Otherwise the sum is calculated in a similar way as was done in the studentAvg but here the row is kept constant at testno-1 (the -1 is to account for indexing based at 0) and the columns vary from 1-12 and the result is added to the sum and then the sum divided by 12 is returned as the average for that specific test.



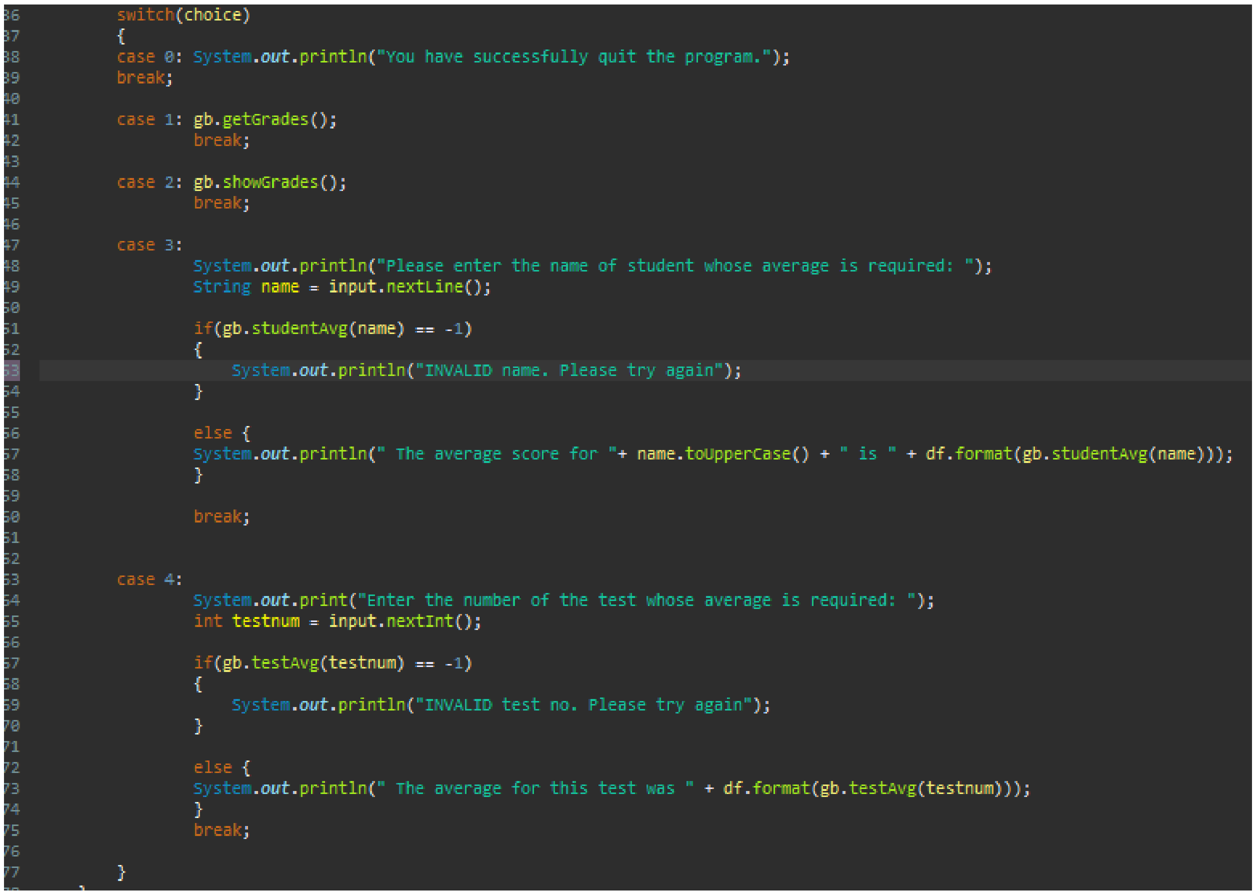
```java
GradeBook gb = new GradeBook();

System.out.println("First, Please enter the names of each student\n");
gb.setName();

System.out.println("Next, Please enter the grades of student\n");
gb.setGrades();
```

Now in my Client Code CourseGrades.java, First I made a gradebook gb based on the object i made. Then the names of the 12 students are entered using the setName() method. Similarly the grades are set using the setGrades() method.



```java
int choice = 1;

while(choice != 0)
{

System.out.println("Please select one of the following options: ");
System.out.println("1. View a Student's Grades");
System.out.println("2. View a Spreadsheet of Class Scores");
System.out.println("3. View a Student's Average Grade");
System.out.println("4. View the Average Grade for a Test");
System.out.println("0. QUIT Program ");

choice = input.nextInt();
    input.nextLine();
```

Then the options that the user can do with the test grades is shown and they are required to input a choice on what they want to select. A while loop is set so the user can keep selecting different options until they want to quit which is when they input 0 for choice causing the while loop to terminate.



```
        switch(choice)
        {
        case 0: System.out.println("You have successfully quit the program.");
        break;

        case 1: gb.getGrades();
                break;

        case 2: gb.showGrades();
                break;

        case 3:
                System.out.println("Please enter the name of student whose average is required: ");
                String name = input.nextLine();

                if(gb.studentAvg(name) == -1)
                {
                    System.out.println("INVALID name. Please try again");
                }

                else {
                System.out.println(" The average score for "+ name.toUpperCase() + " is " + df.format(gb.studentAvg(name)));
                }

                break;


        case 4:
                System.out.print("Enter the number of the test whose average is required: ");
                int testnum = input.nextInt();

                if(gb.testAvg(testnum) == -1)
                {
                    System.out.println("INVALID test no. Please try again");
                }

                else {
                System.out.println(" The average for this test was " + df.format(gb.testAvg(testnum)));
                }
                break;

        }
```

A switch statement is used for the different options:
When user types in 0 a message is shown the that they have successfully quit.

When they press 1 the getGrades method is launched

When they type 2 the showGrades method is launched showing a spreadsheet of the data.

When they press 3 they are asked the name of the student whose average is needed. If the average of the student is -1 then it's recognized that an invalid name was typed in so an error message was shown. Otherwise there is a print message which shows the student's average to 2 decimal places.

Similarly, when they press 4 they calculate the test average. If its -1 then it shows that an invalid test no was inputted. Otherwise the average is displayed.