/\* Chapter 10 test. APCOMPUTERSCIENCE . Read the comments on each line of code and explain or complete if the code is missing. Each comment has a value of 5 points of 100. The last comments has a value of 15 points. Explain your work clearly.

\*/

// Case Study 9.1: Student class.

**public** **class** Student {

 **private** String name;

 **private** tests;

 **public** Student(){

 **this**("");

 }

 **public** Student(String nm){

 **this**(nm, 3);

 }

 **public** Student(String nm, **int** n){

 name = nm;

 tests = **new** **int**[n];

 **for** (**int** i = 0; i < tests.length; i++)

 tests[i] = 0;

 }

 // Name is nm and scores are in t

 **public** Student(String nm, **int**[] t){

 name = nm;

 tests = **new** **int**[t.length];

 **for** (**int** i = 0; i < tests.length; i++)

 tests[i] = t[i];

 }

 // Builds a copy of s

 **public** Student(Student s){

 **this**(s.name, s.tests);

 }

 **public** **void** setName (String nm){

 name = nm;

 }

 **public** String getName (){

 **return** name;

 }

 **public** **void** setScore (**int** i, **int** score){

 tests[i - 1] = score;

 }

 **public** **int** getScore (**int** i){

 **return** tests[i - 1];

 }

 **public** **int** getAverage(){

 **int** sum = 0;

 **for** (**int** score : tests)

 sum += score;

 **return** sum / tests.length;

 }

 **public** **int** getHighScore(){

 **int** highScore = 0;

 **for** (**int** score : tests)

 highScore = Math.*max* (highScore, score);

 **return** highScore;

 }

 **public** String toString(){

 String str = "Name: " + name + "\n";

 **for** (**int** i = 0; i < tests.length; i++)

 str += "test " + i + ": " + tests[i] + "\n";

 str += "Average: " + getAverage();

 **return** str;

 }

 **public** String validateData(){

 **if** (name.equals ("")) **return** "SORRY: name required";

 **for** (**int** score : tests){

 **if** (score < 0 || score > 100){

 String str = "SORRY: must have "+ 0

 + " <= test score <= " + 100;

 **return** str;

 }

 }

 **return** **null**;

 }

}

// Case Study 9.1: TestScoresModel class

**public** **class** TestScoresModel{

  Student[] students; // Array of students

  **int** indexSelectedStudent; // Position of current student

  **int** studentCount; // Current number of students

 **public** TestScoresModel(){

 }

 **public** String add(Student s){

 }

 }

 **public** String replace(Student s){

 **if** (indexSelectedStudent == -1)

 **return** "Must add a student first";

 **else**{

 students[indexSelectedStudent] = s;

 **return** **null**;

 }

 }

 // Navigation methods

 **public** Student first(){

 Student s = **null**;

 **if** (studentCount == 0)

 indexSelectedStudent = -1;

 **else**{

 indexSelectedStudent = 0;

 s = students[indexSelectedStudent];

 }

 **return** s;

 }

 **public** Student previous(){

 Student s = **null**;

 **if** (studentCount == 0)

 indexSelectedStudent = -1;

 **else**{

 indexSelectedStudent

 = Math.*max* (0, indexSelectedStudent - 1);

 s = students[indexSelectedStudent];

 }

 **return** s;

 }

 **public** Student next(){

 Student s = **null**;

 **if** (studentCount == 0)

 indexSelectedStudent = -1;

 **else**{

 indexSelectedStudent

 = Math.*min* (studentCount - 1, indexSelectedStudent + 1);

 s = students[indexSelectedStudent];

 }

 **return** s;

 }

 **public** Student last(){

 Student s = **null**;

 **if** (studentCount == 0)

 indexSelectedStudent = -1;

 **else**{

 indexSelectedStudent = studentCount - 1;

 s = students[indexSelectedStudent];

 }

 **return** s;

 }

 // Accessors to observe data

 **public** Student currentStudent(){

 **if** (indexSelectedStudent == -1)

 **return** **null**;

 **else**

 **return** students[indexSelectedStudent];

 }

 **public** **int** size(){

 **return** studentCount;

 }

 **public** **int** currentPosition(){

 **return** indexSelectedStudent;

 }

 **public** **int** getClassAverage(){

 }

 **public** Student getHighScore(){

 **if** (studentCount == 0)

 **return** **null**;

 **else**{

 Student s = students[0];

 **for** (**int** i = 1; i < studentCount; i++)

 **if** (s.getHighScore() < students[i].getHighScore())

 s = students[i];

 **return** s;

 }

 }

 **public** String toString(){

 String result = "";

 **for** (**int** i = 0; i < studentCount; i++)

 result = result + students[i] + "\n";

 **return** result;

 }

}

// Case Study 9.1: TestScoresView class

**import** java.util.Scanner;

**public** **class** TestScoresView{

 **private** TestScoresModel model;

 **public** TestScoresView(TestScoresModel m){

 model = m;

 run();

 }

 // Menu-driven command loop

 **private** **void** run(){

 **while** (**true**){

 System.*out*.println("Number of students: " + model.size());

 System.*out*.println("Index of current student: " +

 model.currentPosition());

 displayMenu();

 **int** command = getCommand("Enter a number [1-11]: ", 1, 11);

 **if** (command == 11)

 **break**;

 runCommand(command);

 }

 }

 **private** **void** displayMenu(){

 }

 **private** **int** getCommand(String prompt, **int** low, **int** high){

 **int** numero = 0;

 **boolean** num;

 num = **true**;

 System.*out*.print(prompt);

 **while** (num == **true**){

 Scanner reader = **new** Scanner(System.*in*);

 numero = reader.nextInt();

 **if** (numero < 1 || numero > 11)

 {

 System.*out*.println("Enter a valid number: ");

 }

 **else**

 num = **false**;

 }

 System.*out*.print(prompt);

 **return** numero;

 }

 **private** **void** runCommand(**int** command){

 }

}