import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

import java.io.\*;

/\*\*

\* This class elicits student and grade infor from the user and then runs several tests on it to determine the capability

\* of the student to pass the course. It includes how many points are needed, whether it is impossible or guaranteed to pass.

\*

\* @author Michael R. Boykin

\* @version 09/12/2017

\*/

public class Grades

{

// instance variables - replace the example below with your own

Scanner userInput = new Scanner(System.in);

String studentName;

int availablePoints, studentPoints, semesterPoints;

/\*\*

\* The main method where the java compiler will start outside of an API.

\*

\* @param None.

\* @return None.

\*/

public static void main (String[] args)

{

//

ArrayList<String> grades = new ArrayList<String>();

Grades grades1 = new Grades();

grades1.inputData();

grades1.printReport();

//grades1.calculateGrades();

}

/\*\*

\* This method elicits the data that the tests will check to:

\*

\* @param None

\* @return None

\*/

public void inputData ()

{

//Elicits name and grades input from the user.

System.out.print("Enter the student's name:> ");

studentName = userInput.nextLine();

System.out.print("How many points in the semester are available now?:> ");

availablePoints = userInput.nextInt();

System.out.print("How many points has the student earned this semester?:> ");

studentPoints = userInput.nextInt();

System.out.print("How many points are there total in the semester?:> ");

semesterPoints = userInput.nextInt();

System.out.println(); //adds a line space to separate report from inputs.

}

/\*\*

\* This method writes the initial report headeter information and then calls calculateGrades to finish the tests.

\*

\* @param None

\* @return None

\*/

public void printReport ()

{

//

System.out.println("Student Grade Report");

System.out.println();

System.out.println("Student Name: " + studentName);

System.out.println("points possible so far: " + availablePoints);

System.out.println("student points so far: " + studentPoints);

System.out.println("max semester points: " + semesterPoints);

System.out.println("points remaining: " + (semesterPoints - studentPoints));

//System.out.println(" so far so far points semester A B C D F");

//System.out.println();

//System.out.print(studentName + " " + availablePoints + " " + studentPoints + " " + semesterPoints);

calculateGrades (availablePoints, studentPoints, semesterPoints);

System.out.println(); //added a println here in case we want to add multiple student functionality to this.

}

/\*\*

\* This method writes the grade check tests to the screen:

\*

\* @param int availablePoints the points that the course has assigned so far this semester.

\* @param int studentPoints the points that the student has earned so far this semester.

\* @param int semesterPoints the points that the course will be assigned in total this semester.

\* @return None

\*/

public void calculateGrades (int availablePoints, int studentPoints, int semesterPoints)

{

String[] letterGrades = new String[]{"A","B","C","D","F"};

double[] grades = new double[5];

double gradeA = 0.90;

double nextGrade = semesterPoints \* 10.0;

//nextGrade starts with a higher than possible semester grade so that the buildString loop

//below will always evaluate as false when checking if A returns N/A for the first time.

int pointsLeft = semesterPoints - availablePoints;

//pointsLeft is how many points are left for the student to get this semester.

//this loop builds the array for storing grade points required for A, B...F by end of semester.

for(int index = 0; index < 4; index++){

//at start, index = 0 therefore gradeA will be 0.90 and will be calculated

//to the total minimum student points needed by end of semester to ensure grade.

grades[index] = (gradeA - (0.10 \* index)) \* semesterPoints;

}

//because F can be 0 - 59 we have to manually set this value.

grades[4]=0.00;

String returnString;

//buildString for loop

for(int index = 0; index < 5; index++){

//

returnString = "";

System.out.print("To get a grade of " + letterGrades[index] + " is: ");

if((studentPoints + pointsLeft) < grades[index]){returnString = "impossible.";}

if((studentPoints >= grades[index]) && (studentPoints < nextGrade)){returnString = "guaranteed.";}

if((studentPoints >= nextGrade)){returnString = "N/A.";}

//if all the above checks fail to evaluate as true, then the grade is possible.

if((studentPoints < grades[index]) && ((studentPoints + pointsLeft) > grades[index])){

int pointsNeeded = (int) Math.ceil(((double) grades[index] - studentPoints));

returnString += "possible: " + pointsNeeded + "pts needed.";

}

if(grades[index] == 0 && (( (((double)studentPoints + pointsLeft)/semesterPoints)\*100)<60.0)){returnString += " (Fail)";}

nextGrade = grades[index];

System.out.println(returnString);

}

//System.out.println(returnString);

//return returnString;

}

}