```
package ap.compsci.unit.pkg8;
import java.util.Scanner;
//@author lucca
public class APCompsciUnit8 {

   public static void main(String[] args)
   {

      Gradebook instance = new Gradebook();

      boolean continueProgram = true;

      while (continueProgram)
      {
         System.out.println("What would you like to do? Input the integer value associated with
the task that you would like to complete.");
         System.out.println("1: Get the corresponding ID number from the full name of a
student.");
         System.out.println("2: Get the corresponding name of a student from their ID number.");
         System.out.println("3: Get the grade that a student got for a test in a specific class.");
         System.out.println("4: Modify the grade that a student got for a test in a specific class.");
         System.out.println("5: Get a specific's student's GPA.");
         System.out.println("6: Get a specific's student's score for a specific class.");
         System.out.println("7: Get a specific's student's test score matrix.");

         int methodID = 0;

         while (methodID < 1 || methodID > 7)
         {
            Scanner methodIdentifier = new Scanner(System.in);
            methodID = methodIdentifier.nextInt();
            if (methodID < 1 || methodID > 7)
            {
               System.out.println("Error. Task ID is not valid. Please try again.");
            }
         }

         if (methodID == 1)
         {
            System.out.println("You have chosen to get the corresponding ID number from the full
name of a student.");
            System.out.println("Input the full name of the student.");

            int studentID = 999999999;
```

```
            String studentName = null;

            while (studentID > instance.getNumberOfStudents())
            {
                Scanner inputStudentName = new Scanner(System.in);
                studentName = inputStudentName.nextLine();

                studentID = instance.getStudentID(studentName);

                if (studentID > instance.getNumberOfStudents())
                {
                    System.out.println("Student does not exist in database. Please enter a valid
name.");
                }
            }

            System.out.println(studentName + "'s student ID is " + studentID + ".");
        }
        else if (methodID == 2)
        {
            System.out.println("You have chosen to get the corresponding name of a student from
their ID number.");

            System.out.println("Input the ID of the student.");

            int studentID = 999999999;

            while (studentID > instance.getNumberOfStudents())
            {
                Scanner inputStudentID = new Scanner(System.in);
                studentID = inputStudentID.nextInt();

                if (studentID > instance.getNumberOfStudents())
                {
                    System.out.println("Student ID not valid. Please enter a valid ID.");
                }
            }

            String studentName = instance.getStudentName2D(studentID);
            System.out.println("Student ID " + studentID + "'s name is " + studentName + ".");

        }
        else if (methodID == 3)
        {
```

```
            System.out.println("You have chosen to get the grade that a student got for a test in a
specific class.");

            System.out.println("Input the student's ID.");

            int studentID = 999999999;

            while (studentID > instance.getNumberOfStudents())
            {
               Scanner inputStudentID = new Scanner(System.in);
               studentID = inputStudentID.nextInt();

               if (studentID > instance.getNumberOfStudents())
               {
                  System.out.println("Student ID not valid. Please enter a valid ID.");
               }
            }

            System.out.println("Input the integer value (ID) associated with the class that you
would like to access the test data for.");
            System.out.println("0: Mathematics");
            System.out.println("1: English");
            System.out.println("2: Science");
            System.out.println("3: Social Studies");
            System.out.println("4: Language");
            System.out.println("5: Computer Science");
            System.out.println("6: Art");

            int classID = 999999999;

            while (classID < 0 || classID > 6)
            {
               Scanner inputClassID = new Scanner(System.in);
               classID = inputClassID.nextInt();

               if (classID < 0 || classID > 6)
               {
                  System.out.println("Error. Class ID does not correspond to an actual class.
Please try again.");
               }
            }

            System.out.println("Input the test number.");
```

```
            int testNumber = 999999999;

            while (testNumber < 0 || testNumber > 8)
            {
               Scanner inputTestNumber = new Scanner(System.in);
               testNumber = inputTestNumber.nextInt();

               if (testNumber < 0 || testNumber > 8)
               {
                  System.out.println("Error. Test number not valid. Please try again.");
               }

            }

            System.out.println(instance.getGrade(studentID, classID, testNumber));

         }
         else if (methodID == 4)
         {
            System.out.println("You have chosen to change the grade that a student got for a test
in a specific class.");

            System.out.println("Input the student's ID.");

            int studentID = 999999999;

            while (studentID > instance.getNumberOfStudents())
            {
               Scanner inputStudentID = new Scanner(System.in);
               studentID = inputStudentID.nextInt();

               if (studentID > instance.getNumberOfStudents())
               {
                  System.out.println("Student ID not valid. Please enter a valid ID.");
               }
            }

            System.out.println("Input the integer value (ID) associated with the class that you
would like to access the test data for.");
            System.out.println("0: Mathematics");
            System.out.println("1: English");
            System.out.println("2: Science");
            System.out.println("3: Social Studies");
            System.out.println("4: Language");
```

```
            System.out.println("5: Computer Science");
            System.out.println("6: Art");

            int classID = 999999999;

            while (classID < 0 || classID > 6)
            {
               Scanner inputClassID = new Scanner(System.in);
               classID = inputClassID.nextInt();

               if (classID < 0 || classID > 6)
               {
                  System.out.println("Error. Class ID does not correspond to an actual class.
Please try again.");
               }
            }

            System.out.println("Input the test number.");

            int testNumber = 999999999;

            while (testNumber < 0 || testNumber > 8)
            {
               Scanner inputTestNumber = new Scanner(System.in);
               testNumber = inputTestNumber.nextInt();

               if (testNumber < 0 || testNumber > 8)
               {
                  System.out.println("Error. Test number not valid. Please try again.");
               }

            }

            System.out.println("Input the grade the student got (between 0 and 100).");

            int newGrade = 999999999;

            while (newGrade < 0 || newGrade > 100)
            {
               Scanner inputGrade = new Scanner(System.in);
               newGrade = inputGrade.nextInt();

               if (testNumber < 0 || testNumber > 100)
               {
```

```
                System.out.println("Error. Grade number is not valid. Please try again.");
            }

        }

        System.out.println(instance.changeGrade(studentID, classID, testNumber,
newGrade));

    }
    else if (methodID == 5)
    {
        System.out.println("You have chosen to retrieve a student's GPA.");

        System.out.println("Input the student's ID.");

        int studentID = 999999999;

        while (studentID > instance.getNumberOfStudents())
        {
            Scanner inputStudentID = new Scanner(System.in);
            studentID = inputStudentID.nextInt();

            if (studentID > instance.getNumberOfStudents())
            {
                System.out.println("Student ID not valid. Please enter a valid ID.");
            }
        }

        System.out.println(instance.getGPA(studentID));

    }
    else if (methodID == 6)
    {
        System.out.println("You have chosen to change the grade that a student got for a test
in a specific class.");

        System.out.println("Input the student's ID.");

        int studentID = 999999999;

        while (studentID > instance.getNumberOfStudents())
        {
            Scanner inputStudentID = new Scanner(System.in);
            studentID = inputStudentID.nextInt();
```

```
            if (studentID > instance.getNumberOfStudents())
            {
                System.out.println("Student ID not valid. Please enter a valid ID.");
            }
        }

        System.out.println("Input the integer value (ID) associated with the class that you
would like to access the test data for.");
        System.out.println("0: Mathematics");
        System.out.println("1: English");
        System.out.println("2: Science");
        System.out.println("3: Social Studies");
        System.out.println("4: Language");
        System.out.println("5: Computer Science");
        System.out.println("6: Art");

        int classID = 999999999;

        while (classID < 0 || classID > 6)
        {
            Scanner inputClassID = new Scanner(System.in);
            classID = inputClassID.nextInt();

            if (classID < 0 || classID > 6)
            {
                System.out.println("Error. Class ID does not correspond to an actual class.
Please try again.");
            }
        }

        System.out.println(instance.getFinalScore(studentID, classID));

    }
    else
    {
        System.out.println("You have print a student's gradebook in matrix form.");

        System.out.println("Input the student's ID.");

        int studentID = 999999999;

        while (studentID > instance.getNumberOfStudents())
        {
```

```
            Scanner inputStudentID = new Scanner(System.in);
            studentID = inputStudentID.nextInt();

            if (studentID > instance.getNumberOfStudents())
            {
               System.out.println("Student ID not valid. Please enter a valid ID.");
            }
         }

         instance.getGradeMatrix(studentID);

      }

      System.out.println("Would you like to continue utilising the Gradebook Program?");
      System.out.println("Input the number associated with your option.");
      System.out.println("0: NO");
      System.out.println("1: YES");

      int continueProgramInt = 999999999;

      while (continueProgramInt < 0 || continueProgramInt > 1)
      {
         Scanner continueP = new Scanner(System.in);
         continueProgramInt = continueP.nextInt();
         if (continueProgramInt < 0 || continueProgramInt > 1)
         {
            System.out.println("Error. Entry is not valid. Please try again.");
         }
      }

      if(continueProgramInt == 0)
      {
         continueProgram = false;
      }
      else
      {
         continueProgram = true;
      }

   }

}
```

```
package ap.compsci.unit.pkg8;
 //@author lucca
public class Gradebook {

   Student[] allStudents = {new Student("Jake McMillan"), new Student("Jacob Jonis"), new
Student("Aman Parikh"), new Student("Elana Munasinghe"), new Student("Edward Choi"), new
Student("Marc Distel"), new Student("Laijon Best")};

   int testsPerClass = 8;
   int numberOfClasses = 7;
   int[][][] grades = new int[allStudents.length][numberOfClasses][testsPerClass];

   public int getNumberOfStudents()
   {
      return allStudents.length;
   }

   public int getStudentID(String studentName)
   {
      int returnFailsafe = 999999999;
      for(int ID = 0; ID < allStudents.length; ID++)
      {
         if(allStudents[ID].getStudentName().equals(studentName))
         {
            return ID;
         }
      }
      return returnFailsafe;
   }

   public String getStudentName2D(int studentID)
   {
      return allStudents[studentID].getStudentName();
   }

   public String getClassName(int classID)
   {
      if(classID == 0)
      {
         return "Mathematics";
      }
      else if(classID == 1)
      {
         return "English";
```

```
        }
        else if(classID == 2)
        {
            return "Science";
        }
        else if(classID == 3)
        {
            return "Social Studies";
        }
        else if(classID == 4)
        {
            return "Language";
        }
        else if(classID == 5)
        {
            return "Computer Science";
        }
        else if(classID == 6)
        {
            return "Art";
        }
        else
        {
            return "class ID invalid.";
        }

    }

    public String getGrade(int studentID, int classID, int testNumber)
    {
        if(grades[studentID][classID][testNumber] > 0)
        {
            return getStudentName2D(studentID) + "'s " + getClassName(classID) + " grade for test
#" + testNumber + " is" + grades[studentID][classID][testNumber] + ".";
        }
        else
        {
            return getStudentName2D(studentID) + "'s " + getClassName(classID) + " grade for test
#" + testNumber + " is " + grades[studentID][classID][testNumber] + ". They may not have taken
the test yet.";
        }
    }

    public String changeGrade(int studentID, int classID, int testNumber, int newGrade)
```

```
    {
        grades[studentID][classID][testNumber] = newGrade;
        return getStudentName2D(studentID) + "'s " + getClassName(classID) + " grade for test #"
+ testNumber + " has been updated to " + grades[studentID][classID][testNumber];
    }


    public String getGPA(int studentID)
    {
        int GPA = 0;
        int finalPercent = 0;
        int totalGrades = 0;
        int numberOfGrades = 0;
        for(int i = 0; i < grades[studentID].length; i++)
        {
            for(int j = 0; j < grades[studentID][i].length; j++)
            {
                if(grades[studentID][i][j] != 0)
                {
                    totalGrades += grades[studentID][i][j];
                    numberOfGrades++;
                }
            }
        }

        if(numberOfGrades != 0)
        {
            finalPercent = totalGrades / numberOfGrades;
        }
        else
        {
            return getStudentName2D(studentID) + "'s gradebook is empty.";
        }

        if(finalPercent >= 95) //finalPercent is equivilent to an A+
        {
            return getStudentName2D(studentID) + "'s GPA is 4.5";
        }
        else if(finalPercent >= 90) //finalPercent is equivilent to an A
        {
            return getStudentName2D(studentID) + "'s GPA is 4.0";
        }
        else if(finalPercent >= 85) //finalPercent is equivilent to a B+
        {
```

```
            return getStudentName2D(studentID) + "'s GPA is 3.5";
        }
        else if(finalPercent >= 80) //finalPercent is equivilent to a B
        {
            return getStudentName2D(studentID) + "'s GPA is 3.0";
        }
        else if(finalPercent >= 75) //finalPercent is equivilent to a C+
        {
            return getStudentName2D(studentID) + "'s GPA is 2.5";
        }
        else if(finalPercent >= 70) //finalPercent is equivilent to a C
        {
            return getStudentName2D(studentID) + "'s GPA is 2.0";
        }
        else if(finalPercent >= 65) //finalPercent is equivilent to a D+
        {
            return getStudentName2D(studentID) + "'s GPA is 1.5";
        }
        else if(finalPercent >= 60) //finalPercent is equivilent to a D
        {
            return getStudentName2D(studentID) + "'s GPA is 1.0";
        }
        else //finalPercent is equivilent to an F
        {
            return getStudentName2D(studentID) + " will not pass this year of gradeschool if they
continue to get these grades.";
        }
    }

    public String getFinalScore(int studentID, int classID)
    {
        int finalPercent = 0;
        int totalGrades = 0;
        int numberOfGrades = 0;
        for(int i = 0; i < grades[studentID][classID].length; i++)
        {
            if(grades[studentID][classID][i] != 0)
            {
                totalGrades += grades[studentID][classID][i];
                numberOfGrades++;
            }

        }
```

```
        if(numberOfGrades != 0)
        {
            finalPercent = totalGrades / numberOfGrades;
        }
        else
        {
            return getStudentName2D(studentID) + " has no grades in " + getClassName(classID) +
" class.";
        }

        return getStudentName2D(studentID) + "'s final score for " + getClassName(classID) + " is
" + finalPercent + "%.";
    }

    public void getGradeMatrix(int studentID)
    {
        System.out.println("Test Number: " + "       1   2   3   4   5   6   7   8 ");
        System.out.println("Mathematics" + ":      " + getGradeMatrixAssistant(studentID, 0, 0) + " "
+ getGradeMatrixAssistant(studentID, 0, 1) + " " + getGradeMatrixAssistant(studentID, 0, 2) + " "
+ getGradeMatrixAssistant(studentID, 0, 3) + " " + getGradeMatrixAssistant(studentID, 0, 4) + " "
+ getGradeMatrixAssistant(studentID, 0, 5) + " " + getGradeMatrixAssistant(studentID, 0, 6) + " "
+ getGradeMatrixAssistant(studentID, 0, 7));
        System.out.println("English" + ":          " + getGradeMatrixAssistant(studentID, 1, 0) + " " +
getGradeMatrixAssistant(studentID, 1, 1) + " " + getGradeMatrixAssistant(studentID, 1, 2) + " " +
getGradeMatrixAssistant(studentID, 1, 3) + " " + getGradeMatrixAssistant(studentID, 1, 4) + " " +
getGradeMatrixAssistant(studentID, 1, 5) + " " + getGradeMatrixAssistant(studentID, 1, 6) + " " +
getGradeMatrixAssistant(studentID, 1, 7));
        System.out.println("Science" + ":          " + getGradeMatrixAssistant(studentID, 2, 0) + " " +
getGradeMatrixAssistant(studentID, 2, 1) + " " + getGradeMatrixAssistant(studentID, 2, 2) + " " +
getGradeMatrixAssistant(studentID, 2, 3) + " " + getGradeMatrixAssistant(studentID, 2, 4) + " " +
getGradeMatrixAssistant(studentID, 2, 5) + " " + getGradeMatrixAssistant(studentID, 2, 6) + " " +
getGradeMatrixAssistant(studentID, 2, 7));
        System.out.println("Social Studies" + ":   " + getGradeMatrixAssistant(studentID, 3, 0) + " "
+ getGradeMatrixAssistant(studentID, 3, 1) + " " + getGradeMatrixAssistant(studentID, 3, 2) + " "
+ getGradeMatrixAssistant(studentID, 3, 3) + " " + getGradeMatrixAssistant(studentID, 3, 4) + " "
+ getGradeMatrixAssistant(studentID, 3, 5) + " " + getGradeMatrixAssistant(studentID, 3, 6) + " "
+ getGradeMatrixAssistant(studentID, 3, 7));
        System.out.println("Language" + ":         " + getGradeMatrixAssistant(studentID, 4, 0) + " "
+ getGradeMatrixAssistant(studentID, 4, 1) + " " + getGradeMatrixAssistant(studentID, 4, 2) + " "
+ getGradeMatrixAssistant(studentID, 4, 3) + " " + getGradeMatrixAssistant(studentID, 4, 4) + " "
+ getGradeMatrixAssistant(studentID, 4, 5) + " " + getGradeMatrixAssistant(studentID, 4, 6) + " "
+ getGradeMatrixAssistant(studentID, 4, 7));
        System.out.println("Computer Science" + ": " + getGradeMatrixAssistant(studentID, 5, 0) +
" " + getGradeMatrixAssistant(studentID, 5, 1) + " " + getGradeMatrixAssistant(studentID, 5, 2) +
```

```
" " + getGradeMatrixAssistant(studentID, 5, 3) + " " + getGradeMatrixAssistant(studentID, 5, 4) +
" " + getGradeMatrixAssistant(studentID, 5, 5) + " " + getGradeMatrixAssistant(studentID, 5, 6) +
" " + getGradeMatrixAssistant(studentID, 5, 7));
        System.out.println("Art" + ":              " + getGradeMatrixAssistant(studentID, 6, 0) + " " +
getGradeMatrixAssistant(studentID, 6, 1) + " " + getGradeMatrixAssistant(studentID, 6, 2) + " " +
getGradeMatrixAssistant(studentID, 6, 3) + " " + getGradeMatrixAssistant(studentID, 6, 4) + " " +
getGradeMatrixAssistant(studentID, 6, 5) + " " + getGradeMatrixAssistant(studentID, 6, 6) + " " +
getGradeMatrixAssistant(studentID, 6, 7));
    }

    public String getGradeMatrixAssistant(int studentID, int classID, int testNumber)
    {
        int length = String.valueOf(grades[studentID][classID][testNumber]).length();

        if(length == 1)
        {
            return "00" + grades[studentID][classID][testNumber];
        }
        else if(length == 2)
        {
            return "0" + grades[studentID][classID][testNumber];
        }
        else
        {
            String returnStatement = Integer.toString(grades[studentID][classID][testNumber]);
            return returnStatement;
        }
    }

}
```

```
package ap.compsci.unit.pkg8;
 // @author lucca
public class Student {

    String studentName;

    public Student(String name){
        studentName = name;
    }

    public String getStudentName(){
        return studentName;
    }

}
```