```
package grade_book_8;
// Fig. 7.18: GradeBook.java
// Grade book using two-dimensional array to store grades.

public class GradeBook
{
  private String courseName; // name of course this grade book represents
  private int grades[][]; // two-dimensional array of student grades

  // two-argument constructor initializes courseName and grades array
  public GradeBook( String name, int gradesArray[][] )
  {
    courseName = name; // initialize courseName
    grades = gradesArray; // store grades
  } // end two-argument GradeBook constructor

  // method to set the course name
  public void setCourseName( String name )
  {
    courseName = name; // store the course name
  } // end method setCourseName

  // method to retrieve the course name

// method to retrieve the course name
  public String getCourseName()
  {
    return courseName;
  } // end method getCourseName

  // display a welcome message to the GradeBook user
  public void displayMessage()
  {
    // getCourseName gets the name of the course
    System.out.printf( "Welcome to the grade book for\n%s!\n\n",
      getCourseName() );
  } // end method displayMessage

  // perform various operations on the data
  public void processGrades()
  {
    // output grades array
    outputGrades();

    // call methods getMinimum and getMaximum
    System.out.printf( "\n%s %d\n%s %d\n\n",
      "Lowest grade in the grade book is", getMinimum(),
      "Highest grade in the grade book is", getMaximum() );

    // output grade distribution chart of all grades on all tests

   outputBarChart();
  } // end method processGrades

  // find minimum grade
  public int getMinimum()
  {
    // assume first element of grades array is smallest
    int lowGrade = grades[ 0 ][ 0 ];

    // loop through rows of grades array
    for ( int studentGrades[] : grades )
    {
      // loop through columns of current row
      for ( int grade : studentGrades )
      {
        // if grade less than lowGrade, assign it to lowGrade
        if ( grade < lowGrade )
          lowGrade = grade;
      } // end inner for
    } // end outer for

    return lowGrade; // return lowest grade
  } // end method getMinimum

  // find maximum grade

public int getMaximum()
  {
    // assume first element of grades array is largest
    int highGrade = grades[ 0 ][ 0 ];

    // loop through rows of grades array
    for ( int studentGrades[] : grades )
    {
      // loop through columns of current row
      for ( int grade : studentGrades )
      {
        // if grade greater than highGrade, assign it to highGrade
        if ( grade > highGrade )
          highGrade = grade;
      } // end inner for
    } // end outer for

    return highGrade; // return highest grade
  } // end method getMaximum

  // determine average grade for particular student (or set of grades)
  public double getAverage( int setOfGrades[] )
  {
    int total = 0; // initialize total

    // sum grades for one student

  for ( int grade : setOfGrades )
      total += grade;

    // return average of grades
    return (double) total / setOfGrades.length;
  } // end method getAverage

  // output bar chart displaying overall grade distribution
  public void outputBarChart()
  {
    System.out.println( "Overall grade distribution:" );

    // stores frequency of grades in each range of 10 grades
    int frequency[] = new int[ 11 ];

    // for each grade in GradeBook, increment the appropriate frequency
    for ( int studentGrades[] : grades )
    {
      for ( int grade : studentGrades )
        ++frequency[ grade / 10 ];
    } // end outer for

    // for each grade frequency, print bar in chart
    for ( int count = 0; count < frequency.length; count++ )
    {
      // output bar label ( "00-09: ", ..., "90-99: ", "100: " )

if ( count == 10 )
        System.out.printf( "%5d: ", 100 );
      else
        System.out.printf( "%02d-%02d: ",
          count * 10, count * 10 + 9  );

      // print bar of asterisks
      for ( int stars = 0; stars < frequency[ count ]; stars++ )
        System.out.print( "*" );

      System.out.println(); // start a new line of output
    } // end outer for
  } // end method outputBarChart

  // output the contents of the grades array
  public void outputGrades()
  {
    System.out.println( "The grades are:\n" );
    System.out.print( "            " ); // align column heads

    // create a column heading for each of the tests
    for ( int test = 0; test < grades[ 0 ].length; test++ )
      System.out.printf( "Test %d  ", test + 1 );

    System.out.println( "Average" ); // student average column heading

    // create rows/columns of text representing array grades
    for ( int student = 0; student < grades.length; student++ )
    {
      System.out.printf( "Student %2d", student + 1 );

      for ( int test : grades[ student ] ) // output student's grades
        System.out.printf( "%8d", test );

      // call method getAverage to calculate student's average grade;
      // pass row of grades as the argument to getAverage
      double average = getAverage( grades[ student ] );
      System.out.printf( "%9.2f\n", average );
    } // end outer for
  } // end method outputGrades
```

```
} // end class GradeBook
```

```
package grade_book_8;
// Fig. 7.19: GradeBookTest.java
// Creates GradeBook object using a two-dimensional array of grades.

public class GradeBookTest
{
  // main method begins program execution
  public static void main( String args[] )
  {
    // two-dimensional array of student grades
    int gradesArray[][] = { { 87, 96, 70 },
                            { 68, 87, 90 },
                            { 94, 100, 90 },
                            { 100, 81, 82 },
                            { 83, 65, 85 },
                            { 78, 87, 65 },
                            { 85, 75, 83 },
                            { 91, 94, 100 },
                            { 76, 72, 84 },
                            { 87, 93, 73 } };

    GradeBook myGradeBook = new GradeBook(
      "CS101 Introduction to Java Programming", gradesArray );
    myGradeBook.displayMessage();
    myGradeBook.processGrades();
  } // end main
} // end class GradeBookTest
```