Exercise-5

Arrays

1. Rainfall Class

Write a RainFall class that stores the total rainfall for each of 12 months into an array of doubles. The program should have methods that return the following;

- the total rainfall for the year
- the average monthly rainfall
- the month with the max rain
- the month with the least rain

Demonstrate the class in a complete program.

2. Payroll Class

Write a Payroll class that uses the following arrays as fields:

- **employeeId.** An array of seven integers to hold employee identification numbers. The array should be initialized with the following numbers:

  5658845 4520125 7895122 8777541

  8451277 1302850 7580489

- **hours.** An array of seven integers to hold the number of hours worked by each employee
- **payRate.** An array of seven doubles to hold each employees hourly pay rate
- **wages.** An array of seven doubles to hold each employees gross wages

The class should relate the data in each array through the subscripts. For example, the number in element 0 of the hours array should be the number of hours worked by the employee whose identification number is stored in element 0 of the employeeId array. That same employee's pay rate should be stored in element 0 of the payRate array.

In addition to the appropriate accessor and mutator methods, the class should have a method that accepts an employee s identification number as an argument and returns the gross pay for that employee.

Demonstrate the class in a complete program that displays each employee number and asks the user to enter that employee's hours and pay rate. It should then display each employee's identification number and gross wages.

3. Charge Account Validation

Create a class with a method that accepts a charge account number as its argument. The method should determine whether the number is valid by comparing it to the following list of valid charge account numbers:

5658845	4520125	7895122	8777541	8451277	1302850
8080152	4562555	5552012	5050552	7825877	1250255
L00523L	6545231	3852085	7576651	7881200	4581002

These numbers should be stored in an array or an **ArrayList** object. Use a sequential search to locate the number passed as an argument. If the number is in the array, the method should return true, indicating the number is valid. If the number is not in the array, the method should return false, indicating the number is invalid.

Problem Write a program that tests the class by asking the user to enter a charge account number.

the program should display a message indicating whether the number is valid or invalid.

4. Charge Account Modification

Modify the charge account validation class that you wrote for Programming Challenge 3 so it reads the list of valid charge account numbers from a file. Use Notepad or another text editor to create the file.

5. Driver's License Exam

The local Driver's License Office has asked you to write a program that grades the written portion of the driver's license exam. The exam has 20 multiple choice questions. Here are the correct answers:

1. B	6. A	11. B	16. C
2. D	7. B	12. C	17. C
3. A	8. A	13. D	18. B
4. A	9. C	14. A	19. D
5. C	10. D	15. D	20. A

A student must correctly answer 15 of the 20 questions to pass the exam.

Write a class named DriverExam that holds the correct answers to the exam in an array field.

The class should also have an array field that holds the student's answers. The class should have the following methods:

- **passed.** Returns true if the student passed the exam, or false if the student failed
- **totalCorrect.** Returns the total number of correctly answered questions
- **totalIncorrect.** Returns the total number of incorrectly answered questions
- **questionsMissed.** An int array containing the question numbers of the questions that

  the student missed

Demonstrate the class in a complete program that asks the user to enter a student's answers, and then displays the results returned from the **DriverExam** class's methods.

6. Quarterly Sales Statistics

Write a program that lets the user enter four quarterly sales figures for six divisions of a company. The figures should be stored in a two-dimensional array. Once the figures are entered, the program should display the following data for each quarter:

- A list of the sales figures by division
- Each division's increase or decrease from the previous quarter (this will not be displayed for the first quarter)
- The total sales for the quarter
- The company's increase or decrease from the previous quarter (this will not be displayed for the first quarter)
- The average sales for all divisions that quarter
- The division with the highest sales for that quarter

7. Grade Book

A teacher has five students who have taken four tests. The teacher uses the following grading

scale to assign a letter grade to a student, based on the average of his or her four test scores:

Test Score	Letter Grade
90-100	A
80-89	B
70-79	C
60-69	D
0-59	F

Write a class that uses a string array or an **ArrayList** object to hold the five student's names, an array of five characters to hold the five students' letter grades, and five arrays of four doubles each to hold each student's set of test scores. The class should have methods that return a specific student's name, the average test score, and a letter grade based on the average.

Demonstrate the class in a program that allows the user to enter each student's name and his or her four test scores. It should then display each student's average test score and letter grade.

8. Grade Book Modification

Modify the grade book application in Programming Challenge 7 so that it drops each student's lowest score when determining the test score averages and letter grades.

9. Lottery Application

Write a Lottery class that simulates a lottery. The class should have an array of five integers named lotteryNumbcrs. The constructor should use the Random class (from the Java API) to generate a random number in the range of 0 through 9 for each clement in the array. The class should also have a method that accepts an array of five integers that represent a person's lottery picks. The method is to compare the corresponding elements in the two arrays and return the number of digits that match. For example, the following shows the lotteryNumbers array and the user's array with sample numbers stored in each. There are two matching digits (elements 2 and 4).

lotteryNumbers array:

7	**4**	**9**	**1**	**3**

User's array :

4	**2**	**9**	**7**	**3**

In addition, the class should have a method that returns a copy of the lotteryNumbers array.

Demonstrate the class in a program that asks the user to enter five numbers. The program should display the number of digits that match the randomly generated lottery numbers. If all of the digits match, display a message proclaiming the user a grand prize winner.

10. Array Operations

Write a program with an array that is initialized with test data. Use any primitive data type of your choice. The program should also have the following methods:

• getTotal. This method should accept, a one-dimensional array as its argument and return the total of the values in the array.

• getAverage. This method should accept a one-dimensional array as its argument and return the average of the values in the array.

• getHighest. This method should accept a one-dimensional array as its argument and return the highest value in the array.

• getLowest. This method should accept a one-dimensional array is us argument and return the lowest value in the array.

Demonstrate each of the methods in the program.

11. 2D Array Operations

Write a program that creates a two-dimensional array initialized with test data. Use any primitive data type that you wish. The program should have the following methods:

• **getTotal.** This method should accept a two-dimensional array as its argument and

return the total of all the values in the array.

• **getAverage.** This method should accept a two-dimensional array as its argument and

return the average of all the values in the array.

• **getRowTotal.** This method should accept a two-dimensional array as its first argument

and an integer as its second argument. The second argument should be the subscript

of a row1 in the array. The method should return the total of the values in the

specified row.

• **getColumnTotal.** This method should accept a two-dimensional array as its first argument

and an integer as its second argument. The second argument should he the subscript

of a column in the array. The method should return the total of the values in the

specified column.

• **getHighestInRow.** This method should accept a two-dimensional array as its first

argument and an integer as us second argument. The second argument should be the

subscript of a row in the array. The method should return the highest value in the

specified row of the array,

• **getLowestInRow,** This method should accept a two-dimensional array as its first argument

and an integer as its second argument. The second argument should be the subscript of a row in the array. The method should return the lowest value in the specified row of the array.

Demonstrate each of the methods in this program.

12. Phone Book ArrayList

Write a class named PhoneBookEntry that has fields for a person's name and phone number. The class should have a constructor and appropriate accessor and mutator methods. Then write a program that creates at least five PhoneBookEntry objects

and stores them in an ArrayList. Use a loop to display the contents of each object in the ArrayList.

13. Trivia Game

In this programming challenge, you will create a simple trivia game for two players. The program will work like this;

- Starting with player 1, each player gets a turn at answering 5 trivia questions. (There are 10 questions, 5 for each player.) When a question is displayed, four possible answers are also displayed. Only one of the answers is correct, and if the player selects the correct answer, he or she earns a point.
- After answers have been selected for all of the questions, the program displays the number of points earned by each player and declares the player with the highest number of points the winner.

You are to design a Question class to hold the data for a trivia question. The Question class should have String fields for the following data:

- A trivia question
- Possible answer 1
- Possible answer 2
- Possible answer 3
- Possible answer *4*
- The number of the correct answer (1,2, 3, or 4)

The Question class should have appropriate constructors), accessor, and mutator methods. The program should create an array of 10 Question objects, one for each trivia question. (If you prefer, you can use an ArrayList instead of an array.) Make up your own trivia questions on the subject or subjects of your choice for the objects.