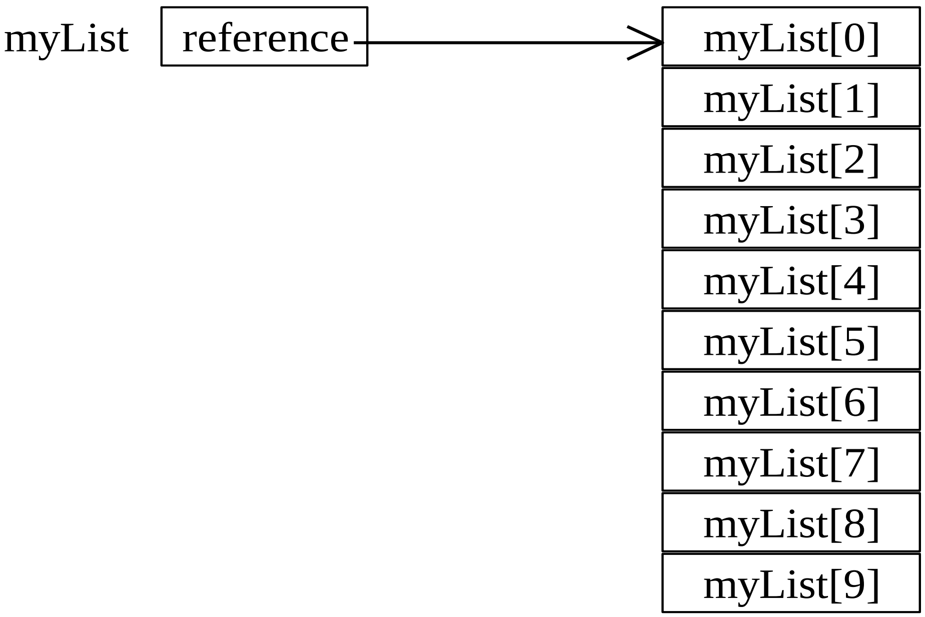
B.Sc. In Software Development. Year 3.
Applications Programming.
Arrays.

LIMERICK INSTITUTE
OF TECHNOLOGY
SCHOOL OF SCIENCE,
ENGINEERING & I.T.
Department of Information Technology

Introducing Arrays

Array is a data structure that represents a collection of the same types of data. Java treats these arrays as objects.

```
double[] myList = new double[10];
```



myList

reference

myList[0]

myList[1]

myList[2]

myList[3]

myList[4]

myList[5]

myList[6]

myList[7]

myList[8]

myList[9]

An Array of 10 Elements
of type `double`

Declaring Array Variables

- `datatype[] arrayname;`

Example:

```
int[] myList;
```

- `datatype arrayname[];`

Example:

```
int myList[];
```

Creating Arrays

```
arrayName = new datatype[arraySize];
```

Example:

```
myList = new double[10];
```

Declaring and Creating in One Step

-
  ```
  datatype[] arrayname =
                 new datatype[arraySize];
  ```

  ```
  double[] myList = new double[10];
  ```

-
  ```
  datatype arrayname[] = new
    datatype[arraySize];
  ```

  ```
  double myList[] = new double[10];
  ```

Initialising Arrays

- Using a loop:

```
for (int i = 0; i < myList.length; i++)
   myList[i] = (double)i;
```

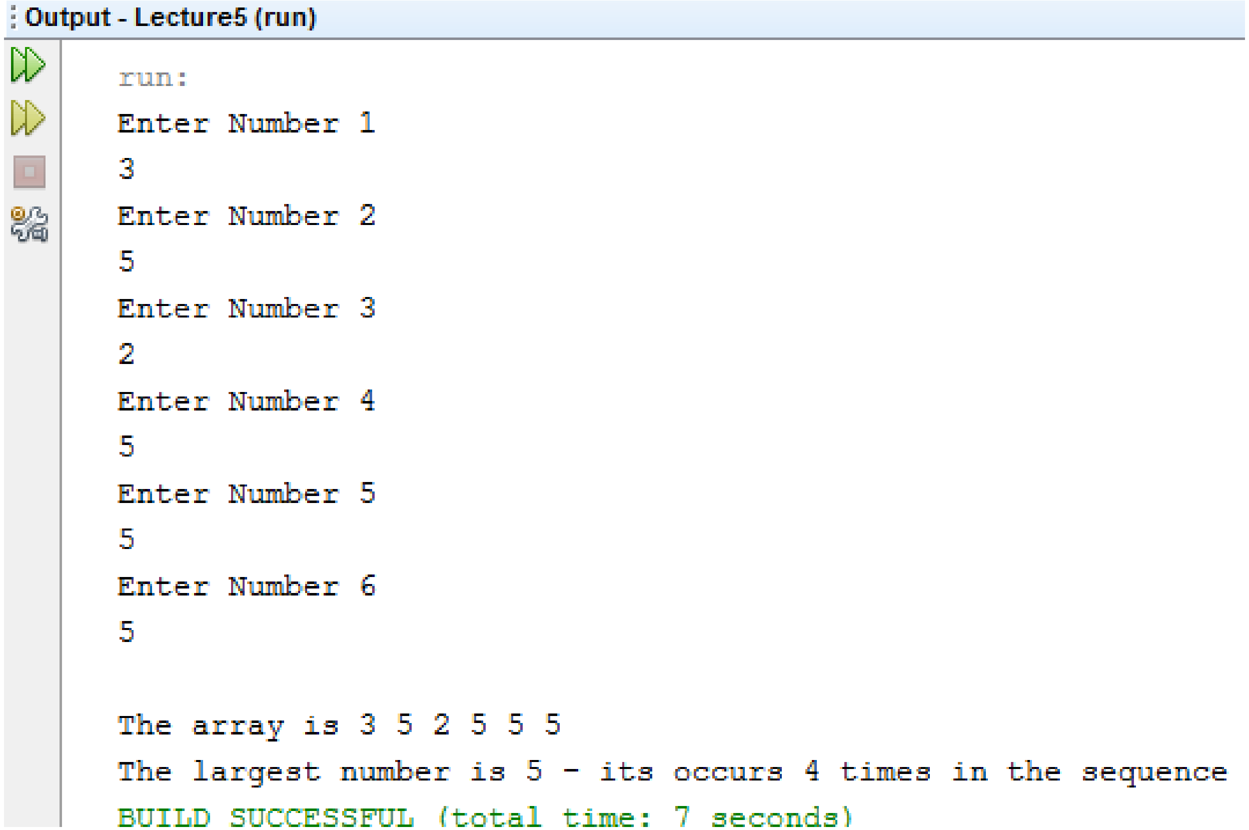
- Declaring, creating, initializing in one step:

```
double[] myList = {1.9, 2.9, 3.4, 3.5};
```

Example 1- TestArray.java

Objective: The program receives 6 numbers as input and finds the largest number as well as counting the occurrence of the largest number entered from the keyboard.

Sample input/output.



```
run:
Enter Number 1
3
Enter Number 2
5
Enter Number 3
2
Enter Number 4
5
Enter Number 5
5
Enter Number 6
5

The array is 3 5 2 5 5 5
The largest number is 5 - its occurs 4 times in the sequence
BUILD SUCCESSFUL (total time: 7 seconds)
```

Example 2 – AssignGrade.java

- **Objective**: read student a number of student scores from the user and then assign grades based on the following scheme:

If the score >= 85 then the grade is

‘A’.

If the scores <= 84 and the score >= 70 then the grade is

‘B‘.

If the score <= 69 and the score >= 50 then the grade is

‘C‘.

If the score <= 49 and the score >= 40 then the grade is

‘D‘.

If the score is <40 then the grade is

‘F‘.

Multidimensional Arrays

```
int[][] matrix = new int[10][10];
```

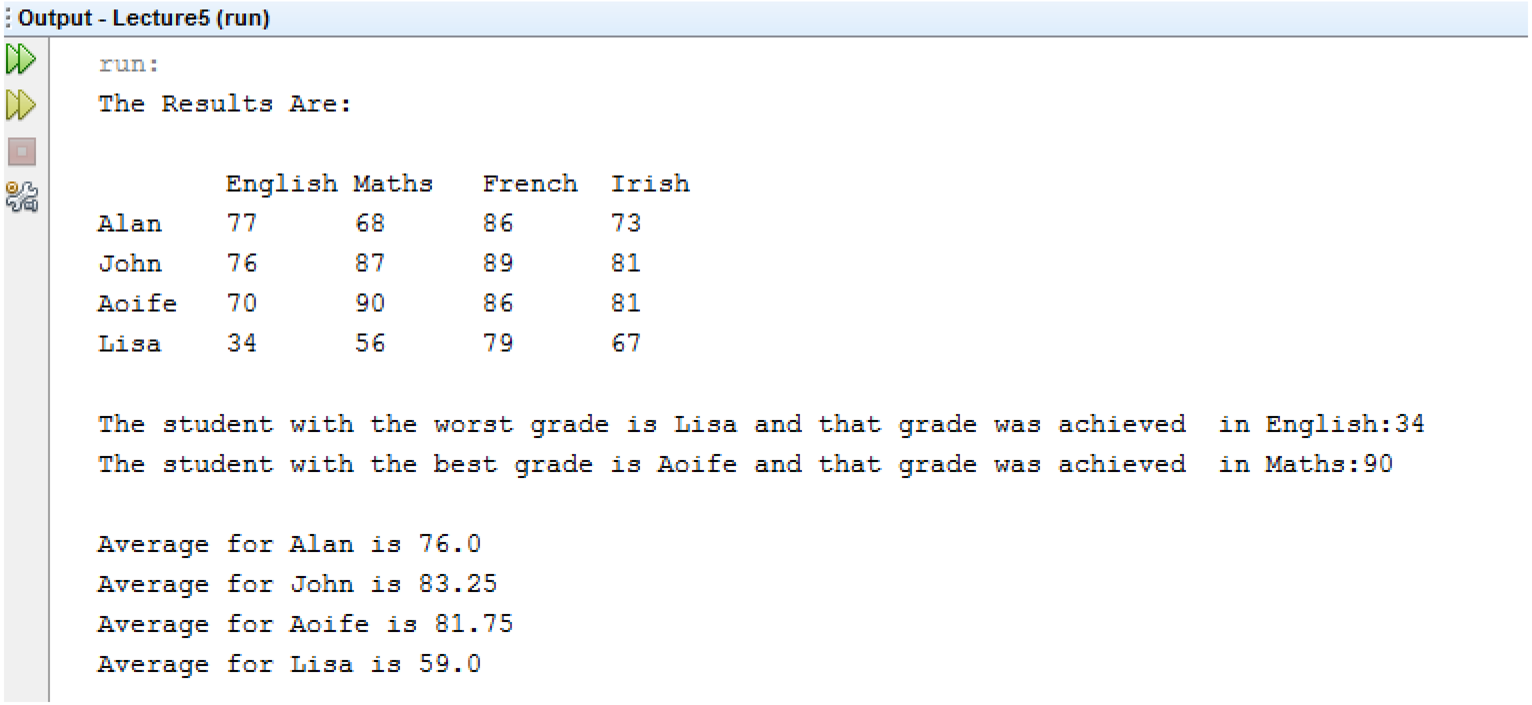
or

```
int matrix[][] = new int[10][10];

for (int i=0; i<matrix.length; i++)
  for (int j=0; j<matrix[i].length; j++)
  {
    matrix[i][j] = (int)(Math.random()*1000);
  }
```

Example 3 – DoubleArrayApplication.java

- **Objective**: Using a Multidimensional array, process a set of student results in four subjects as per the output below:



```
run:
The Results Are:

        English Maths   French  Irish
Alan    77      68      86      73
John    76      87      89      81
Aoife   70      90      86      81
Lisa    34      56      79      67

The student with the worst grade is Lisa and that grade was achieved  in English:34
The student with the best grade is Aoife and that grade was achieved  in Maths:90

Average for Alan is 76.0
Average for John is 83.25
Average for Aoife is 81.75
Average for Lisa is 59.0
```

Ragged Arrays

Each row in a two-dimensional array is itself an array. So, the rows can have different lengths. Such an array is known as *a ragged array*. For example,

```
int[][] matrix =
{
  {1, 2, 3, 4, 5},
  {2, 3, 4, 5},
  {3, 4, 5},
  {4, 5},
  {5}
};
```

Exercise 1

Write a program that creates an array with 100 elements.
You must initialise each element in the array to a random value in the range 1 - 1000;
Your program must then calculate and display (using a method to achieve each of the following tasks)

1. The largest number in the array.
2. The smallest number in the array.
3. The sum of all the numbers.
4. The average of all the numbers.
5. The frequency of the number 7 in the array.

Exercise 2

Write a program that creates an array with 5000 elements.

Each element in this array must be initialised with a random value in the range 0 – 29.

Your program must then output the frequency of each number in the range 0 - 29 which is stored in the array.

You must also output the modal value (the modal value is the value which occurs most frequently in the array.

Exercise 3

Write a program that creates an array with 5000 elements.

Each element in this array must be initialised with a random value in the range 5 – 50.

Your program must firstly output the arrays contents in columns of 20 values.

You must then output the numbers in the array which are multiples of 6, 7 and 8 (if any) along with the index of the array in which they appear.

If there are no multiples of 6, 7 and 8 in the array, your program must print a suitable message.

Exercise 4

Create a 2D array with 20 rows and 10 columns.

Then using an appropriate nested loop structure, initialise each number in the array to a random value in the range 0-30.

Ensure that a single value cannot appear more than once in each row.

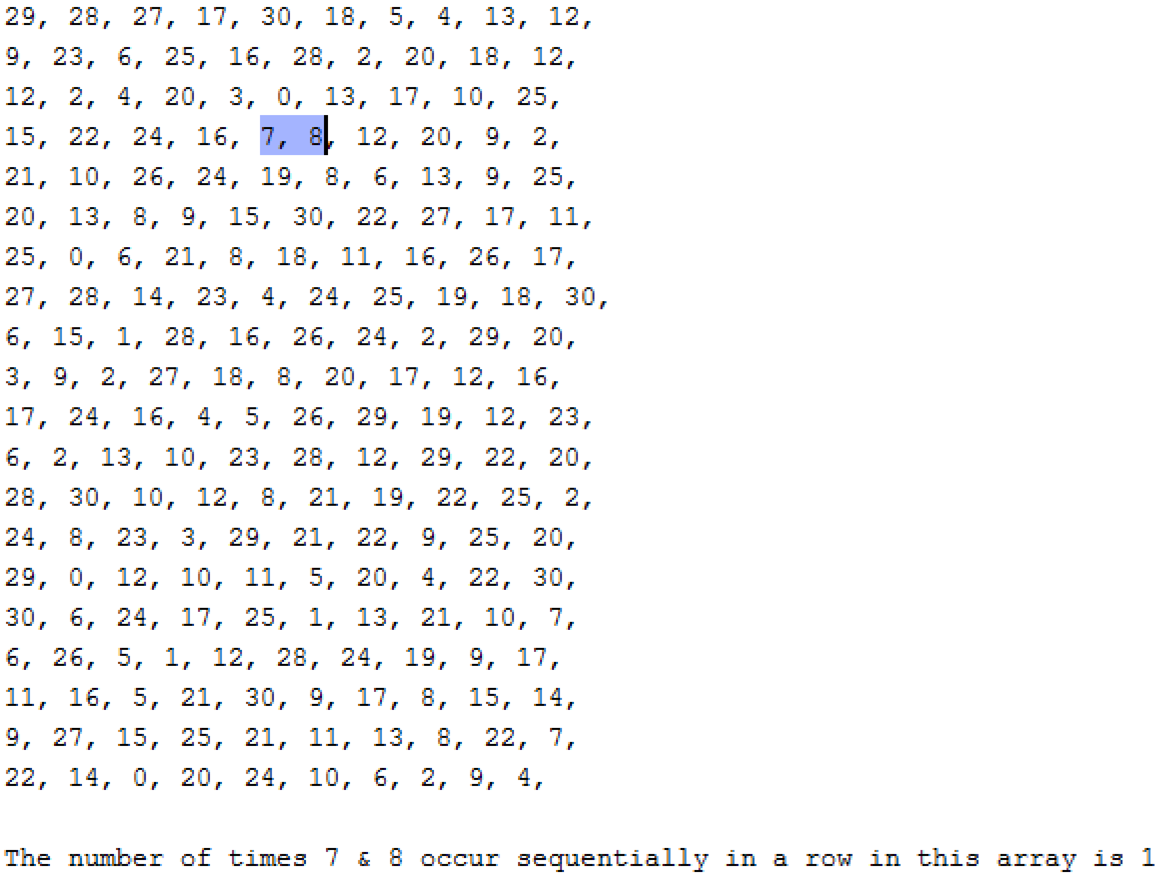
In other words, each row should have values which are randomly generated but must occur only once (in each row).

Display the contents of the entire array (row by row) on the screen.

Determine and output how many times (if any) the numbers 7 and 8 appear sequentially in any row of the array.

Exercise 4

The output should look something like this:

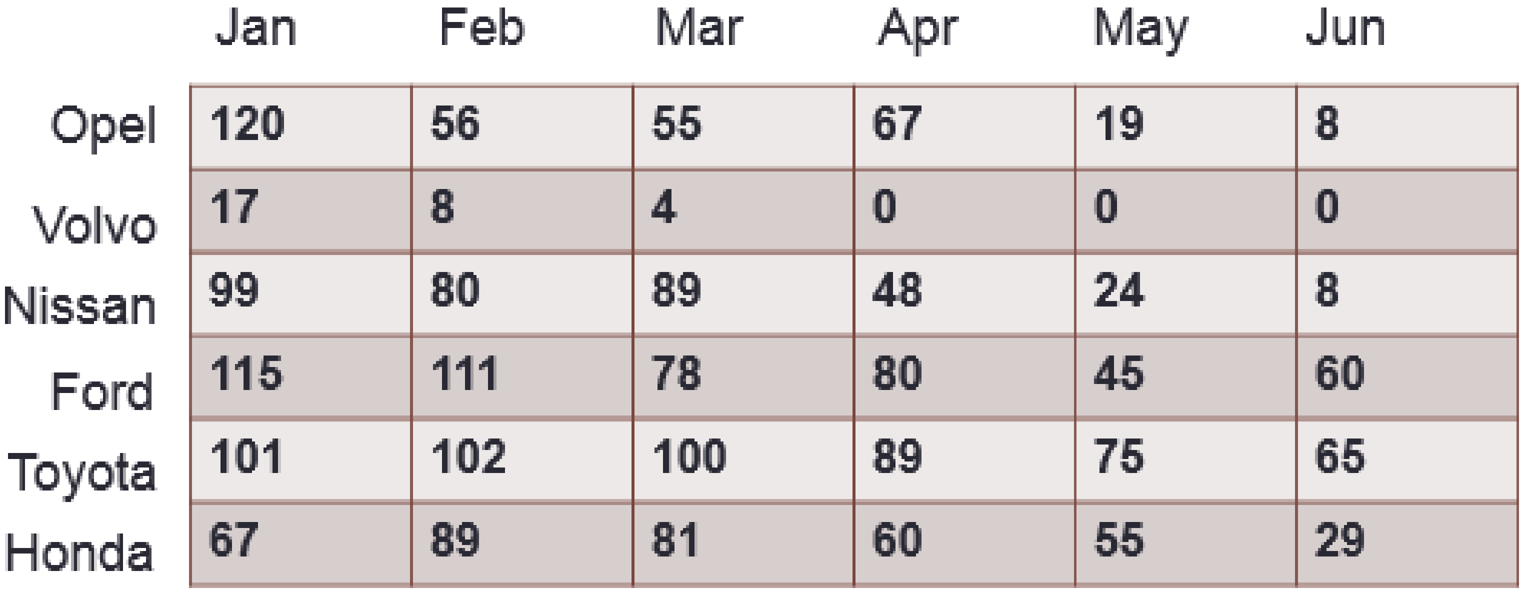


```
29, 28, 27, 17, 30, 18, 5, 4, 13, 12,
9, 23, 6, 25, 16, 28, 2, 20, 18, 12,
12, 2, 4, 20, 3, 0, 13, 17, 10, 25,
15, 22, 24, 16, 7, 8, 12, 20, 9, 2,
21, 10, 26, 24, 19, 8, 6, 13, 9, 25,
20, 13, 8, 9, 15, 30, 22, 27, 17, 11,
25, 0, 6, 21, 8, 18, 11, 16, 26, 17,
27, 28, 14, 23, 4, 24, 25, 19, 18, 30,
6, 15, 1, 28, 16, 26, 24, 2, 29, 20,
3, 9, 2, 27, 18, 8, 20, 17, 12, 16,
17, 24, 16, 4, 5, 26, 29, 19, 12, 23,
6, 2, 13, 10, 23, 28, 12, 29, 22, 20,
28, 30, 10, 12, 8, 21, 19, 22, 25, 2,
24, 8, 23, 3, 29, 21, 22, 9, 25, 20,
29, 0, 12, 10, 11, 5, 20, 4, 22, 30,
30, 6, 24, 17, 25, 1, 13, 21, 10, 7,
6, 26, 5, 1, 12, 28, 24, 19, 9, 17,
11, 16, 5, 21, 30, 9, 17, 8, 15, 14,
9, 27, 15, 25, 21, 11, 13, 8, 22, 7,
22, 14, 0, 20, 24, 10, 6, 2, 9, 4,

The number of times 7 & 8 occur sequentially in a row in this array is 1
```

Exercise 5

The following table represents the number of cars sold in a large local car dealership for the first six months of 2017.



	Jan	Feb	Mar	Apr	May	Jun
Opel	120	56	55	67	19	8
Volvo	17	8	4	0	0	0
Nissan	99	80	89	48	24	8
Ford	115	111	78	80	45	60
Toyota	101	102	100	89	75	65
Honda	67	89	81	60	55	29

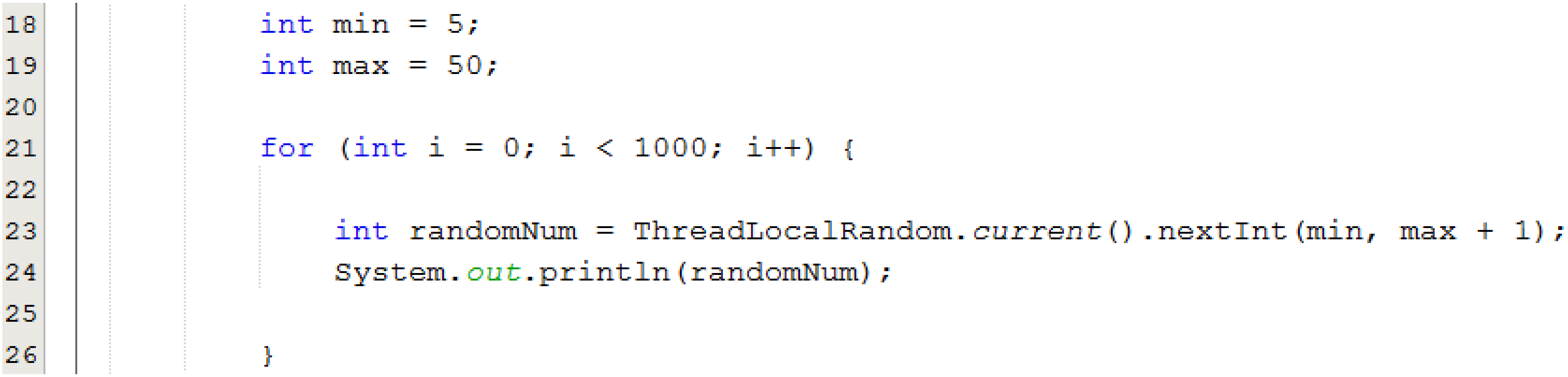
Exercise 5

Write a program that will represent this data, and then calculate and output the following:

1. The total number cars sold in the six month period.
2. The total and average number of each car brand sold over the six month period (outputting the car brand and the number of cars).
3. The best selling car over the six month period.
4. The worst selling car over the six month period.
5. The total number of cars sold for each month.
6. If (for any car brand) there was a month on month increase in the number of cars sold - output the car brand and the months in question. For example, there was an increase in the number of Opel's sold from March to April.

Generating Random Numbers

There are a number of ways in which you can generate random numbers. Here’s one.



```
int min = 5;
int max = 50;

for (int i = 0; i < 1000; i++) {

    int randomNum = ThreadLocalRandom.current().nextInt(min, max + 1);
    System.out.println(randomNum);

}
```

References

Joel Murach 2015, *Murach's Beginning Java with NetBeans* Mike Murach & Associates.
ISBN-13 9781890774844 (Link)

Paul Deitel 2017, *Java How To Program (Early Objects) (11th Edition).* Prentice Hall.
ISBN-13 9780134800271 (Link)

Daniel Liang 2017, *Introduction to Java Programming and Data Structures, Comprehensive Version (11th Edition). Pearson.*
ISBN-13 978-0134670942 (Link)