



COMPUTER SCIENCE DEPARTMENT FACULTY OF
ENGINEERING AND TECHNOLOGY
ADVANCED PROGRAMMING COMP231

Instructor :Murad Njoum
Office : Masri322



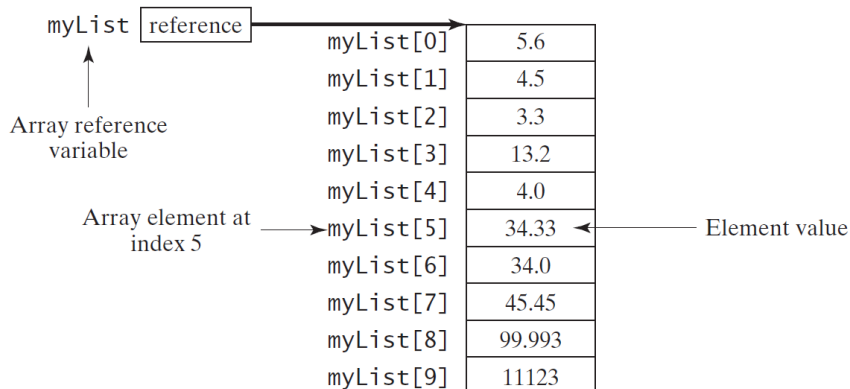
Chapter 7+8 Arrays

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Introducing Arrays

Array is a data structure that represents a collection of the same types of data.

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



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Declaring Array Variables

- `datatype[] arrayRefVar;`

Example:

```
double[] myList;
```

- `datatype arrayRefVar[];` // This style is allowed, but not preferred

Example:

```
double myList[];
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Creating Arrays

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

Example:

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

`myList[0]` references the first element in the array.

`myList[9]` references the last element in the array.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Declaring and Creating in One Step

- `datatype[] arrayRefVar = new datatype[arraySize];`
`double[] myList = new double[10];`
- `datatype arrayRefVar[] = new datatype[arraySize];`
`double myList[] = new double[10];`



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

The Length of an Array

Once an array is created, its size is fixed. It cannot be changed. You can find its size using

`arrayRefVar.length`

For example,

`myList.length` returns 10



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Default Values

When an array is created, its elements are assigned the default value of

0 for the numeric primitive data types,
'\u0000' for char types, and
false for boolean types.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Indexed Variables

The array elements are accessed through the index. The array indices are *0-based*, i.e., it starts from 0 to `arrayRefVar.length-1`. In the example in Figure 6.1, `myList` holds ten double values and the indices are from 0 to 9.

Each element in the array is represented using the following syntax, known as an *indexed variable*:

```
arrayRefVar[index];
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Using Indexed Variables

After an array is created, an indexed variable can be used in the same way as a regular variable. For example, the following code adds the value in `myList[0]` and `myList[1]` to `myList[2]`.

```
myList[2] = myList[0] + myList[1];
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Declaring, creating, initializing Using the Shorthand Notation

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

This shorthand notation is equivalent to the following statements:

```
double[] myList = new double[4];
```

```
myList[0] = 1.9;
```

```
myList[1] = 2.9;
```

```
myList[2] = 3.4;
```

```
myList[3] = 3.5;
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

CAUTION

Using the shorthand notation, you have to declare, create, and initialize the array **all in one statement**.

Splitting it would cause a syntax error. For example, the following is wrong:

```
double[] myList;
```

```
myList = {1.9, 2.9, 3.4, 3.5};
```

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Trace Program with Arrays

Declare array variable values, create an array, and assign its reference to values

```
public class Test {  
    public static void main(String[] args) {  
        int[] values = new int[5];  
        for (int i = 1; i < 5; i++) {  
            values[i] = i + values[i-1];  
        }  
        values[0] = values[1] + values[4];  
    }  
}
```

After the array is created

0	0
1	0
2	0
3	0
4	0



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Initializing arrays with input values

```
Scanner input = new java.util.Scanner(System.in);  
System.out.print("Enter " + myList.length + " values: ");
```

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

Initializing arrays with random values

```
for (int i = 0; i < myList.length; i++) {  
    myList[i] = Math.random() * 100;  
}
```

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Printing arrays

```
for (int i = 0; i < myList.length; i++) {  
    System.out.print(myList[i] + " ");  
}
```

Summing all elements

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

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Finding the largest element

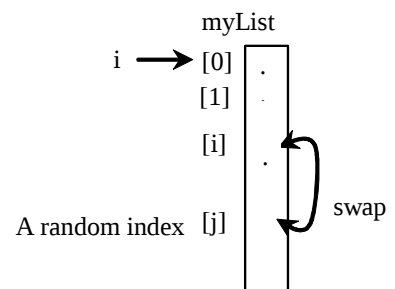
```
double max = myList[0];  
for (int i = 1; i < myList.length; i++) {  
    if (myList[i] > max)  
        max = myList[i];  
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Random shuffling

```
for (int i = 0; i < myList.length - 1; i++) {  
    // Generate an index j randomly  
    int j = (int) (Math.random()  
        * myList.length);  
  
    // Swap myList[i] with myList[j]  
    double temp = myList[i];  
    myList[i] = myList[j];  
    myList[j] = temp;  
}
```



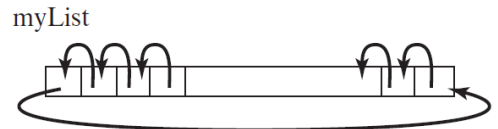
liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Shifting Elements

```
double temp = myList[0]; // Retain the first element
```

```
// Shift elements left  
for (int i = 1; i < myList.length; i++) {  
    myList[i - 1] = myList[i];  
}
```

```
// Move the first element to fill in the last position  
myList[myList.length - 1] = temp;
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Enhanced for Loop (for-each loop)

JDK 1.5 introduced a new for loop that enables you to traverse the complete array sequentially without using an index variable. For example, the following code displays all elements in the array myList:

```
for (double value: myList)  
    System.out.println(value);
```

In general, the syntax is

```
for (elementType value: arrayRefVar) {  
    // Process the value  
}
```

You still have to use an index variable if you wish to traverse the array in a different order or change the elements in the array.

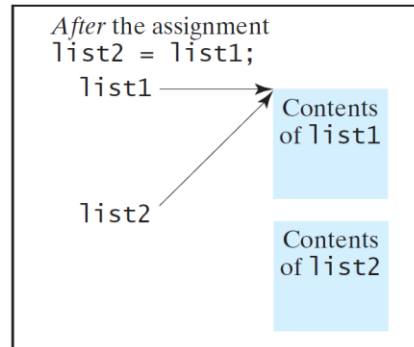
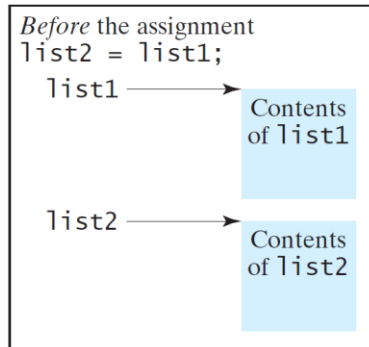


liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Copying Arrays

Often, in a program, you need to duplicate an array or a part of an array. In such cases you could attempt to use the assignment statement (=), as follows:

```
list2 = list1;
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun



Copying Arrays

Using a loop:

```
int[] sourceArray = {2, 3, 1, 5, 10};
```

```
int[] targetArray = new  
    int[sourceArray.length];
```

```
for (int i = 0; i < sourceArrays.length; i++)  
    targetArray[i] = sourceArray[i];
```

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun



The arraycopyUtility:

System.arraycopy(sourceArray, src_pos, targetArray, tar_pos, length);

Example:

```
System.arraycopy(sourceArray, 0, targetArray, 0,  
sourceArray.length);
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Passing Arrays to Methods

```
public static void printArray(int[] array) {  
    for (int i = 0; i < array.length; i++) {  
        System.out.print(array[i] + " ");  
    }  
}
```

Invoke the method

```
int[] list = {3, 1, 2, 6, 4, 2};  
printArray(list);
```

Invoke the method

```
printArray(new int[]{3, 1, 2, 6, 4, 2});
```

Anonymous array



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Anonymous Array

The statement

```
printArray(new int[]{3, 1, 2, 6, 4, 2});
```

creates an array using the following syntax:

```
new dataType[]{literal0, literal1, ..., literalK};
```

There is no explicit reference variable for the array.
Such array is called an *anonymous array*.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Pass By Value

Java uses *pass by value* to pass arguments to a method. There are important differences between passing a value of variables of primitive data types and passing arrays.

- For a parameter of a primitive type value, the actual value is passed. Changing the value of the local parameter inside the method does not affect the value of the variable outside the method.
- For a parameter of an array type, the value of the parameter contains a reference to an array; this reference is passed to the method. Any changes to the array that occur inside the method body will affect the original array that was passed as the argument.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Simple Example

```
public class Test {  
    public static void main(String[] args) {  
        int x = 1; // x represents an int value  
        int[] y = new int[10]; // y represents an array of int values  
  
        m(x, y); // Invoke m with arguments x and y  
  
        System.out.println("x is " + x);  
        System.out.println("y[0] is " + y[0]);  
    }  
  
    public static void m(int number, int[] numbers) {  
        number = 1001; // Assign a new value to number  
        numbers[0] = 5555; // Assign a new value to numbers[0]  
    }  
}
```

x is 1
y[0] is 5555

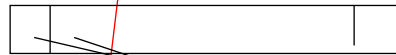


liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

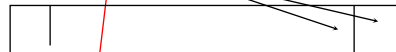
Returning an Array from a Method

```
public static int[] reverse(int[] list) {  
    int[] result = new int[list.length];  
  
    for (int i = 0, j = result.length - 1;  
         i < list.length; i++, j--) {  
        result[j] = list[i];  
    }  
  
    return result;  
}
```

list



result



```
int[] list1 = {1, 2, 3, 4, 5, 6};  
int[] list2 = reverse(list1);
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Trace the reverse Method

```
int[] list1 = {1, 2, 3, 4, 5, 6};  
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {  
    int[] result = new int[list.length];  
    for (int i = 0, j = result.length - 1;  
         i < list.length; i++, j--) {  
        result[j] = list[i];  
    }  
    return result;  
}
```

Declare result and create array

list

1	2	3	4	5	6
---	---	---	---	---	---

result

0	0	0	0	0	0
---	---	---	---	---	---

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njourn



Trace the reverse Method, cont.

```
int[] list1 = {1, 2, 3, 4, 5, 6};  
int[] list2 = reverse(list1);
```

```
public static int[] reverse(int[] list) {  
    int[] result = new int[list.length];  
    for (int i = 0, j = result.length - 1;  
         i < list.length; i++, j--) {  
        result[j] = list[i];  
    }  
    return result;  
}
```

After this, i becomes 6 and
j becomes -1

list

1	2	3	4	5	6
---	---	---	---	---	---

result

6	5	4	3	2	1
---	---	---	---	---	---

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njourn



Searching Arrays

Searching is the process of looking for a specific element in an array; for example, discovering whether a certain score is included in a list of scores. Searching is a common task in computer programming. There are many algorithms and data structures devoted to searching. In this section, two commonly used approaches are discussed, *linear search* and *binary search*.

```
public class LinearSearch {  
    /** The method for finding a key in the list */  
    public static int linearSearch(int[] list, int key) {  
        for (int i = 0; i < list.length; i++)  
            if (key == list[i])  
                return i;  
        return -1;  
    }  
}
```

list [0] [1] [2] ...
key Compare key with list[i] for i = 0, 1, ...



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njourn

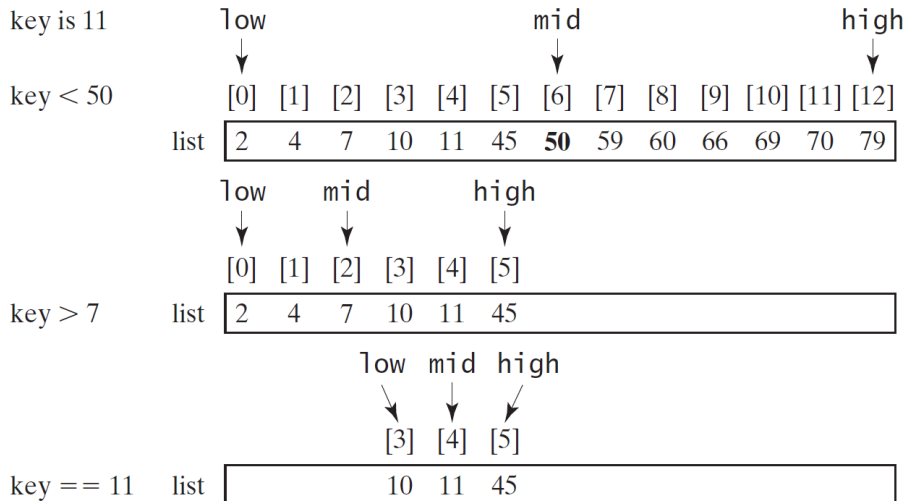
Linear Search

The linear search approach compares the key element, key, *sequentially* with each element in the array list. The method continues to do so until the key matches an element in the list or the list is exhausted without a match being found. If a match is made, the linear search returns the index of the element in the array that matches the key. If no match is found, the search returns -1.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njourn

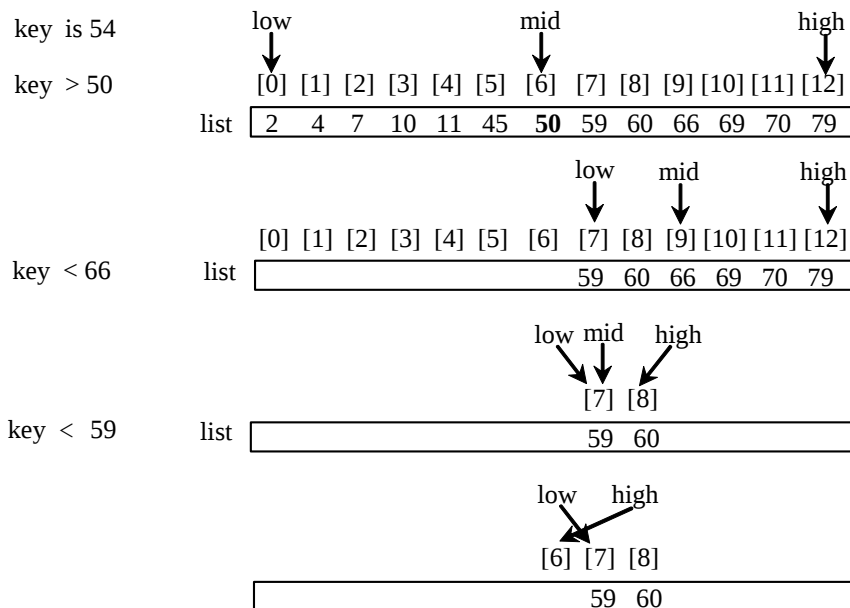
Binary Search, cont.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Binary Search, cont.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Binary Search, cont.

The `binarySearch` method returns the index of the element in the list that matches the search key if it is contained in the list. Otherwise, it returns

-insertion point - 1.

The insertion point is the point at which the key would be inserted into the list.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

From Idea to Solution

```
/** Use binary search to find the key in the list */
public static int binarySearch(int[] list, int key) {
    int low = 0;
    int high = list.length - 1;

    while (high >= low) {
        int mid = (low + high) / 2;
        if (key < list[mid])
            high = mid - 1;
        else if (key == list[mid])
            return mid;
        else
            low = mid + 1;
    }

    return -1 - low;
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

The Arrays.binarySearch Method

Since binary search is frequently used in programming, Java provides several overloaded `binarySearch` methods for searching a key in an array of `int`, `double`, `char`, `short`, `long`, and `float` in the `java.util.Arrays` class. For example, the following code searches the keys in an array of numbers and an array of characters.

```
int[] list = {2, 4, 7, 10, 11, 45, 50, 59, 60, 66, 69, 70, 79};
```

```
System.out.println("Index is " +
```

```
java.util.Arrays.binarySearch(list, 11));
```

Return is 4

```
char[] chars = {'a', 'c', 'g', 'x', 'y', 'z'};
```

```
System.out.println("Index is " +
```

```
java.util.Arrays.binarySearch(chars, 't'));
```

Return is -4 (insertion point is 3, so return is -3-1)

For the `binarySearch` method to work, the array must be pre-sorted in increasing order.

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Sorting Arrays

Sorting, like searching, is also a common task in computer programming. Many different algorithms have been developed for sorting. This section introduces a simple, intuitive sorting algorithms: *selection sort*.

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Command-Line Parameters

```
class TestMain {  
    public static void main(String[] args) {  
        ...  
    }  
}
```

```
java TestMain arg0 arg1 arg2 ... argn
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Main Method Is Just a Regular Method

You can call a regular method by passing actual parameters. Can you pass arguments to main? Of course, yes. For example, the main method in class B is invoked by a method in A, as shown below:

```
public class A {  
    public static void main(String[] args) {  
        String[] strings = {"New York",  
                            "Boston", "Atlanta"};  
        B.main(strings);  
    }  
}  
  
class B {  
    public static void main(String[] args) {  
        for (int i = 0; i < args.length; i++)  
            System.out.println(args[i]);  
    }  
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Processing Command-Line Parameters

In the main method, get the arguments from `args[0]`, `args[1]`, ..., `args[n]`, which corresponds to `arg0`, `arg1`, ..., `argn` in the command line.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Problem: Calculator

- Objective: Write a program that will perform binary operations on integers. The program receives three parameters: an operator and two integers.

Calculator

Run

```
java Calculator 2 + 3
```

```
java Calculator 2 - 3
```

```
java Calculator 2 / 3
```

```
java Calculator 2 . 3
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

2 Dimensional array

```
double[][] distances = {
    {0, 983, 787, 714, 1375, 967, 1087},
    {983, 0, 214, 1102, 1763, 1723, 1842},
    {787, 214, 0, 888, 1549, 1548, 1627},
    {714, 1102, 888, 0, 661, 781, 810},
    {1375, 1763, 1549, 661, 0, 1426, 1187},
    {967, 1723, 1548, 781, 1426, 0, 239},
    {1087, 1842, 1627, 810, 1187, 239, 0},
};
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Declare/Create Two-dimensional Arrays

```
// Declare array ref var
dataType[][] refVar;

// Create array and assign its reference to variable
refVar = new dataType[10][10];

// Combine declaration and creation in one statement
dataType[][] refVar = new dataType[10][10];

// Alternative syntax
dataType refVar[][] = new dataType[10][10];
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Declaring Variables of Two-dimensional Arrays and Creating Two-dimensional Arrays

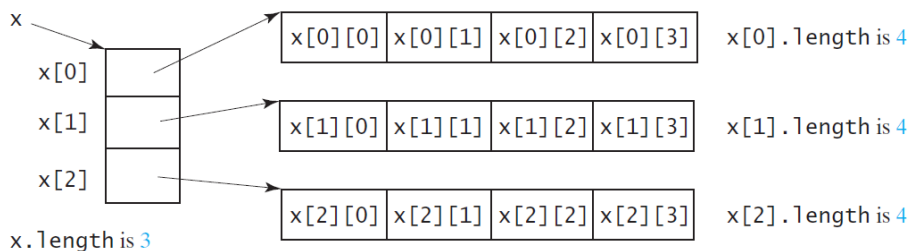
```
int[][] matrix = new int[10][10];  
or  
int matrix[][] = new int[10][10];  
matrix[0][0] = 3;  
  
for (int i = 0; i < matrix.length; i++)  
    for (int j = 0; j < matrix[i].length; j++)  
        matrix[i][j] = (int) (Math.random() * 1000);  
  
double[][] x;
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Lengths of Two-dimensional Arrays

```
int[][] x = new int[3][4];
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Lengths of Two-dimensional Arrays, cont.

```
int[][] array = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9},  
    {10, 11, 12}  
};
```

array.length
array[0].length
array[1].length
array[2].length
array[3].length

array[4].length **ArrayIndexOutOfBoundsException**



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

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,

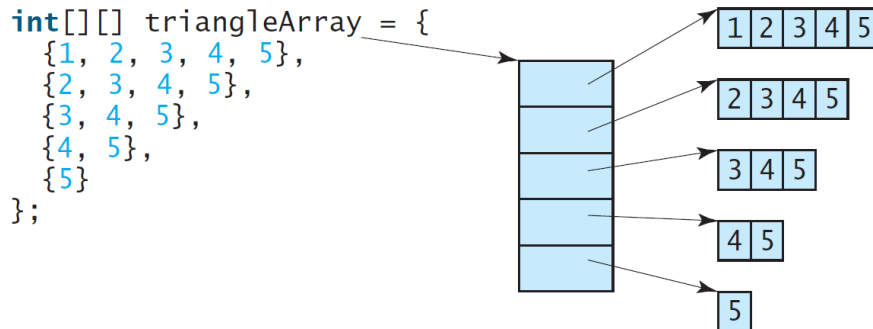
```
int[][] matrix = {  
    {1, 2, 3, 4, 5},  
    {2, 3, 4, 5},  
    {3, 4, 5},  
    {4, 5},  
    {5}  
};
```

matrix.length is 5
matrix[0].length is 5
matrix[1].length is 4
matrix[2].length is 3
matrix[3].length is 2
matrix[4].length is 1



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Ragged Arrays, cont.



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Initializing arrays with input values

```
Scanner input = new Scanner(System.in);  
System.out.println("Enter " + matrix.length + " rows and " +  
    matrix[0].length + " columns: ");  
  
for (int row = 0; row < matrix.length; row++) {  
    for (int column = 0; column < matrix[row].length; column++) {  
        matrix[row][column] = input.nextInt();  
    }  
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Printing arrays

```
for (int row = 0; row < matrix.length; row++) {  
    for (int column = 0; column < matrix[row].length; column++) {  
        System.out.print(matrix[row][column] + " ");  
    }  
  
    System.out.println();  
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Summing all elements

```
int total = 0;  
for (int row = 0; row < matrix.length; row++) {  
    for (int column = 0; column < matrix[row].length; column++) {  
        total += matrix[row][column];  
    }  
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Summing elements by column

```
for (int column = 0; column < matrix[0].length; column++) {  
    int total = 0;  
    for (int row = 0; row < matrix.length; row++)  
        total += matrix[row][column];  
    System.out.println("Sum for column " + column + " is "  
        + total);  
}
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

Problem: Grading Multiple-Choice Test

Students' answer

	0	1	2	3	4	5	6	7	8	9
Student 0	A	B	A	C	C	D	E	E	A	D
Student 1	D	B	A	B	C	A	E	E	A	D
Student 2	E	D	D	A	C	B	E	E	A	D
Student 3	C	B	A	E	D	C	E	E	A	D
Student 4	A	B	D	C	C	D	E	E	A	D
Student 5	B	B	E	C	C	D	E	E	A	D
Student 6	B	B	A	C	C	D	E	E	A	D
Student 7	E	B	E	C	C	D	E	E	A	D

Objective: write a program that grades multiple-choice test.

Key to the Questions:

0 1 2 3 4 5 6 7 8 9
Key D B D C C D A E A D



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun

```

public class GradeExam {
    /** Main method */
    public static void main(String[] args) {
        // Students' answers to the questions
        char[][] answers = {
            {'A', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
            {'D', 'B', 'A', 'B', 'C', 'A', 'E', 'E', 'A', 'D'},
            {'E', 'D', 'D', 'A', 'C', 'B', 'E', 'E', 'A', 'D'},
            {'C', 'B', 'A', 'E', 'D', 'C', 'E', 'E', 'A', 'D'},
            {'A', 'B', 'D', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
            {'B', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
            {'B', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
            {'E', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'};

        // Key to the questions
        char[] keys = {'D', 'B', 'D', 'C', 'C', 'D', 'A', 'E', 'A', 'D'};

        // Grade all answers
        for (int i = 0; i < answers.length; i++) {
            // Grade one student
            int correctCount = 0;
            for (int j = 0; j < answers[i].length; j++) {
                if (answers[i][j] == keys[j])
                    correctCount++;
            }

            System.out.println("Student " + i + "'s correct count is " +
                               correctCount);
        }
    }
}

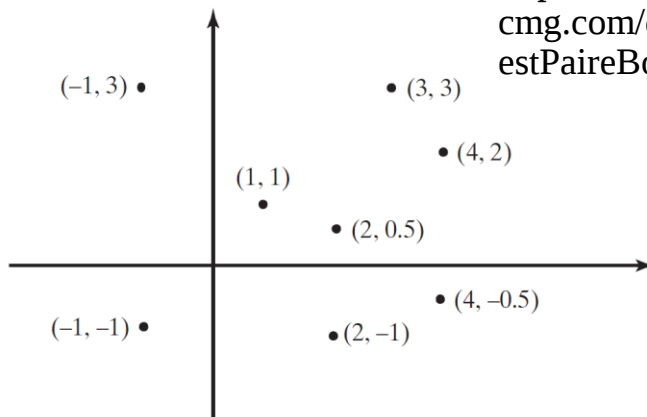
```



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

Problem: Finding Two Points Nearest to Each Other

<https://liveexample.pearsoncmg.com/dsanimation/ClosestPairBook.html>



	x	y
0	-1	3
1	-1	-1
2	1	1
3	2	0.5
4	2	-1
5	3	3
6	4	2
7	4	-0.5



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum

```
import java.util.Scanner;
```

```
public class FindNearestPoints {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter the number of points: ");  
        int numberOfPoints = input.nextInt();  
  
        // Create an array to store points  
        double[][] points = new double[numberOfPoints][2];  
        System.out.print("Enter " + numberOfPoints + " points: ");  
        for (int i = 0; i < points.length; i++) {  
            points[i][0] = input.nextDouble();  
            points[i][1] = input.nextDouble();  
        }  
  
        // p1 and p2 are the indices in the points array  
        int p1 = 0, p2 = 1; // Initial two points  
        double shortestDistance = distance(points[p1][0], points[p1][1],  
            points[p2][0], points[p2][1]); // Initialize shortestDistance  
  
        // Compute distance for every two points  
        for (int i = 0; i < points.length; i++) {  
            for (int j = i + 1; j < points.length; j++) {  
                double distance = distance(points[i][0], points[i][1],  
                    points[j][0], points[j][1]); // Find distance  
            }  
        }  
    }  
}
```

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



```
        if (shortestDistance > distance) {  
            p1 = i; // Update p1  
            p2 = j; // Update p2  
            shortestDistance = distance; // Update  
        }  
    }  
}  
  
// Display result  
System.out.println("The closest two points are " +  
    "(" + points[p1][0] + ", " + points[p1][1] + ") and  
    "(" + points[p2][0] + ", " + points[p2][1] + ")");  
}  
  
/** Compute the distance between two points (x1, y1) and  
(x2, y2)*/  
public static double distance(  
    double x1, double y1, double x2, double y2) {  
    return Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) *  
        (y2 - y1));  
}
```

liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoum



Multidimensional Arrays

```
double[][][] scores = {  
    {{7.5, 20.5}, {9.0, 22.5}, {15, 33.5}, {13, 21.5}, {15, 2.5}},  
    {{4.5, 21.5}, {9.0, 22.5}, {15, 34.5}, {12, 20.5}, {14, 9.5}},  
    {{6.5, 30.5}, {9.4, 10.5}, {11, 33.5}, {11, 23.5}, {10, 2.5}},  
    {{6.5, 23.5}, {9.4, 32.5}, {13, 34.5}, {11, 20.5}, {16, 7.5}},  
    {{8.5, 26.5}, {9.4, 52.5}, {13, 36.5}, {13, 24.5}, {16, 2.5}},  
    {{9.5, 20.5}, {9.4, 42.5}, {13, 31.5}, {12, 20.5}, {16, 6.5}}  
};
```

Which student Which exam Multiple-choice or essay

↓ ↓ ↓

scores[i] [j] [k]



liang introduction to java programming 11th edition ,2019 , Edit By : Mr.Murad Njoun