**[CSE 1310](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/index.html) -** [**Assignments**](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/index.html) **- Programming Assignment 7**

The assignment will be graded out of 100 points.

Some tasks ask you to write code, and specify what name to use for the file where you save that code. You need to use exactly the name that is given (do not change the case, or make any other modification). Remember, the name of the main class must match the filename.

For some tasks you need to answer questions. Create a text document entitled answers.txt, or answers.docx, or answers.pdf, and put all your answers there. Acceptable file formats are plain text, Word document, OpenOffice document, and PDF. Put your name and UTA ID in the file on the first line.

Each task below will instruct you where to put your answers.

**Task 1 (10 pts.)**

public class task1

{

public static void print\_int\_matrix(String name, int[] a)

{

System.out.printf("%7s: ", name);

for (int i = 0; i < a.length; i++)

{

System.out.printf("%7d", a[i]);

}

System.out.printf("\n");

}

public static void foo(int[] x, int[] y)

{

for (int i = 0; i < x.length; i++)

{

x[i] = 100 + x[i];

}

int[] q = {1, 2, 3};

y = q;

}

public static void main(String[] args)

{

int[] a = {3, 2, 5};

int[] b = a;

int[] c = {3, 2, 5};

int[] d = c;

foo(b, d);

print\_int\_matrix("a", a);

print\_int\_matrix("c", c);

}

}

If you execute this program, what will be printed? Put your answer in your answers file.

**Task 2 (10 pts.)**

import java.util.\*;

public class task2

{

public static void print\_string\_arraylist(ArrayList<String> a)

{

for (int i = 0; i < a.size(); i++)

{

System.out.printf("%10s", a.get(i));

}

System.out.printf("\n");

}

public static void foo(ArrayList<String> x, ArrayList<String> y)

{

for (int i = 0; i < x.size(); i++)

{

String current = x.get(i);

if (current.equals("tx"))

{

x.set(i, "Texas");

}

}

y.add("hello");

y.add("world");

y = new ArrayList<String>();

y.add("goodbye");

}

public static void main(String[] args)

{

ArrayList<String> a = new ArrayList<String>();

ArrayList<String> b = a;

ArrayList<String> c = new ArrayList<String>();

ArrayList<String> d = c;

a.add("fort");

a.add("worth");

a.add("tx");

c.add("june");

c.add("july");

c.add("august");

foo(b, d);

print\_string\_arraylist(a);

print\_string\_arraylist(c);

}

}

If you execute this program, what will be printed? Put your answer in your answers file.

**Task 3 (10 pts.)**

File [task3.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task3.java) contains an incomplete program. The goal of the program is to compute the position-wise maxima of two arrays. Complete that program, by defining an array\_max function, that satisfies the following specs:

* Function array\_max takes two arguments, called A, B. They are both arrays of double numbers.
* If the two arrays do NOT have the same length, function array\_max should return null.
* Otherwise, the function should return an array called result, with length equal to the length of A, such that the value at position i of result is the larger among the values at position i of A and B.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the array\_max function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a: 3.2 2.1 5.3 8.0 4.9 5.7

b: 1.1 2.2 3.3 4.4 5.5 6.6

result: 3.2 2.2 5.3 8.0 5.5 6.6

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Task 4 (10 pts.)**

File [task4.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task4.java) contains an incomplete program. The goal of the program is to compute the position-wise maxima of two array lists. Complete that program, by defining an array\_list\_max function, that satisfies the following specs:

* Function array\_list\_max takes two arguments, called A, B. They are both array lists of double numbers.
* If the two array lists do NOT have the same length, function array\_list\_max should return null.
* Otherwise, the function should return an array list called result, with length equal to the length of A, such that the value at position i of result is the larger among the values at position i of A and B.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the array\_list\_max function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a: 3.2 2.1 5.3 8.0 4.9 5.7

b: 1.1 2.2 3.3 4.4 5.5 6.6

result: 3.2 2.2 5.3 8.0 5.5 6.6

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Task 5 (10 pts.)**

File [task5.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task5.java) contains an incomplete program. The goal of the program is to compute the position-wise squares of the values of an array. Complete that program, by defining an array\_square function, that satisfies the following specs:

* Function array\_square takes one argument, called A, that is an array of double numbers.
* The function should return an array called result, with length equal to the length of A, such that the value at position i of result is square of the value at position i of A.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the array\_square function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a: 3.20 2.10 5.30 8.00 4.90 5.70

array\_square(a): 10.24 4.41 28.09 64.00 24.01 32.49

b: 1.10 2.20 3.30 4.40 5.50 6.60

array\_square(b): 1.21 4.84 10.89 19.36 30.25 43.56

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Task 6 (10 pts.)**

File [task6.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task6.java) contains an incomplete program. The goal of the program is to compute the position-wise squares of the values of an array list. Complete that program, by defining an array\_list\_square function, that satisfies the following specs:

* Function array\_list\_square takes one argument, called A, that is an array list of double numbers.
* The function should return an array list called result, with length equal to the length of A, such that the value at position i of result is square of the value at position i of A.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the array\_list\_square function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a: 3.20 2.10 5.30 8.00 4.90 5.70

array\_list\_square(a): 10.24 4.41 28.09 64.00 24.01 32.49

b: 1.10 2.20 3.30 4.40 5.50 6.60

array\_list\_square(b): 1.21 4.84 10.89 19.36 30.25 43.56

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Task 7 (10 pts.)**

File [task7.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task7.java) contains an incomplete program. The goal of the program is to merge the values of two String arrays into a single array. Complete that program, by defining an array\_merge function, that satisfies the following specs:

* Function array\_merge takes two arguments, called A, B, that are arrays of strings.
* The function should return a String array called result, with length equal to the length of A plus the length of B, that contains the values of A, in the order in which they appear in A, followed by the values of B, in the order in which they appear in B.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the array\_merge function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a: Chicago New York Dallas

b: Berlin London Paris Rome

result: Chicago New York Dallas Berlin London Paris Rome

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Task 8 (10 pts.)**

File [task8.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task8.java) contains an incomplete program. The goal of the program is to merge the values of two String array lists into a single array list. Complete that program, by defining an array\_list\_merge function, that satisfies the following specs:

* Function array\_list\_merge takes two arguments, called A, B, that are array lists of strings.
* The function should return an array list of strings called result, with length equal to the length of A plus the length of B, that contains the values of A, in the order in which they appear in A, followed by the values of B, in the order in which they appear in B.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the array\_list\_merge function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a: Chicago New York Dallas

b: Berlin London Paris Rome

result: Chicago New York Dallas Berlin London Paris Rome

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Task 9 (10 pts.)**

File [task9.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task9.java) contains an incomplete program. The goal of the program is to compute the position-wise minima of two matrices (a matrix is a two-dimensional array). Complete that program, by defining a matrix\_min function, that satisfies the following specs:

* Function matrix\_min takes two arguments, called A, B. They are both 2D arrays of double numbers.
* If the two arrays do NOT have the same size (i.e., equal rows, and equal columns), function matrix\_min should return null.
* Otherwise, the function should return a 2D array called result, with rows and columns equal to those of A, such that the value at position (i, j) of the result is the smaller among the values at position (i, j) of A and B.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the matrix\_min function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a:

3.2 2.1 5.3

8.0 4.9 5.7

b:

1.1 2.2 3.3

4.4 5.5 6.6

result:

1.1 2.1 3.3

4.4 4.9 5.7

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Task 10 (10 pts.)**

File [task10.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment7/task10.java) contains an incomplete program. The goal of the program is to compute the average of values in specific regions of a matrix. Complete that program, by defining a matrix\_region\_average function, that satisfies the following specs:

* Function matrix\_region\_average takes five arguments, called A, top, bottom, left, right. A is a 2D array of double numbers. Arguments top, bottom, left, right are all integers.
* The function should return the average of values in all positions (i, j) such that top <= i <= bottom and left <= j <= right.

**IMPORTANT: You are NOT allowed to modify in any way the main function, or any other function that is already provided.**.. You are allowed to modify the provided code by writing the matrix\_region\_average function, and you are also free to write any additional functions that you may find useful.

The complete program should produce this output:

a:

3.2 2.1 5.3

8.0 4.9 5.7

18.0 14.9 15.7

28.0 24.9 25.7

38.0 34.9 35.7

matrix\_region\_average(a, 0, 3, 1, 2) returned 12.40.

matrix\_region\_average(a, 1, 4, 0, 1) returned 21.45.

Your program's output should match EXACTLY the format shown above. There should be no deviations, no extra spaces or lines, no extra punctuation in your output. What you see above as uppercase letters should remain uppercase in your output, what you see as lowercase letters should remain as lowercase in your output, what you see as spaces and punctuation should remain exactly as spaces and punctuation in your output.

**Suggestions**

Pay close attention to all specifications on this page, including file names and submission format. Even in cases where the program works correctly, points will be taken off for non-compliance with the instructions given on this page (such as wrong file names, wrong compression format for the submitted code, and so on). The reason is that non-compliance with the instructions makes the grading process significantly (and unnecessarily) more time consuming. Contact the instructor or TA if you have any questions.

**How to submit**

The assignment should be submitted via [Blackboard](http://elearn.uta.edu). Submit a ZIPPED directory called assignment7.zip (no other forms of compression accepted, contact the instructor or TA if you do not know how to produce .zip files).

To create a zipped directory called assignment7.zip, follow these steps:

1. Create a folder called assignment7.
2. Copy to that folder all your solutions (your answers file, and all your Java files).
3. Zip that folder. On windows, you can zip a folder by right-clicking on the folder, and then selecting Send to->Compressed (zipped) folder.

Your zip file should contain only 9 files: your answers document and all the Java code files (task3.java, task4.java, task5.java, task6.java, task7.java, task8.java, task9.java, task10.java).

**Submission checklist**

* Did you create the answers file with your name, UTA ID, and answers to non-programming tasks?
* Did you zip everything into a file called assignment7.zip?