**[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 8**

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 double[][] foo(double[] x)

{

double[][] result = new double[2][];

result[0] = x;

result[1] = x;

for (int i = 0; i < x.length / 2; i++)

{

double temp = result[0][i];

result[0][i] = result[0][x.length-i-1];

result[0][x.length-i-1] = temp;

}

return result;

}

public static void print\_double\_matrix(double[][] a)

{

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

{

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

{

System.out.printf("%7.1f", a[i][j]);

}

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

}

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

}

public static void print\_double\_array(double[] a)

{

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

{

System.out.printf("%7.1f", a[i]);

}

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

}

public static void main(String[] args)

{

double[] a = {1.1, 2.2, 3.3, 4.4};

double[][] b = foo(a);

print\_double\_array(a);

print\_double\_matrix(b);

}

}

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 foo(ArrayList<String> a, String b)

{

a.add(a.size()/2, b.substring(0, b.length()/2));

}

public static void main(String[] args)

{

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

foo(x, "Dallas");

foo(x, "Chicago");

foo(x, "New York");

foo(x, "Denver");

System.out.println(x);

}

}

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

**Task 3 (10 pts.)**

import java.util.\*;

public class task3

{

public static ArrayList<ArrayList<String>> foo(String b)

{

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

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

{

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

result.add(current);

for (int j = 0; j <= i; j++)

{

current.add(b.substring(0, j+1));

}

}

return result;

}

public static void main(String[] args)

{

ArrayList<ArrayList<String>> x = foo("Texas");

System.out.println(x);

}

}

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

**Task 4 (10 pts.)**

File [task4.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment8/task4.java) contains an incomplete program. The goal of the program is to:

1. Create an empty array list of integers.
2. Insert into that array list some integers, one by one.
3. Insertions should be done in such a way that, after each insertion, the numbers in the array list are sorted in ascending order.

Complete that program, by defining an insert\_before\_larger function, that satisfies the following specs:

* Function insert\_before\_larger takes two arguments, called A, x. Argument A is an array list of integers, and argument x is an integer.
* The function should look for the first element of A that is larger than x, and insert x right before that element.
* For example, if A contains elements 10, 40, 50, in that order, and x = 20, then x should be inserted at position 1, right before element 40. After the insertion, A should contain elements 10, 20, 40, 50, in that order.

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

The complete program should produce this output:

[40]

[10, 40]

[10, 40, 50]

[10, 20, 40, 50]

[10, 20, 30, 40, 50]

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/assignment8/task5.java) contains an incomplete program. The goal of the program is to sort an array list of integers, so that the integers appear in ascending order. Complete that program, by defining a sort\_array function, that satisfies the following specs:

* Function sort\_array takes one argument, called A, that is an array list of integers.
* The function should return an array list called result, with length equal to the length of A. The result should contain the same elements as A, but in ascending order.

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

Hints: my solution also includes (and uses) the insert\_before\_larger function from the previous task. My solution takes five lines of code, in addition to the insert\_before\_larger function. All you have to do is:

* Create a result array list.
* Go through every element of A, and insert it to the result using the insert\_before\_larger function.

The complete program should produce this output:

Input: [40, 10, 50, 20, 30]

Output: [10, 20, 30, 40, 50]

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/assignment8/task6.java) contains an incomplete program. The goal of the program is to sort an array list of integers, so that the integers appear in ascending order. Complete that program, by defining a sort\_array2 function, that satisfies the following specs:

* Function sort\_array2 takes one argument, called A, that is an array list of integers.
* The function MODIFIES A, so that it contains the same elements as before, but in ascending order.

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

Hints: my solution also includes (and uses) the insert\_before\_larger and sort\_array functions from the previous tasks. My solution takes four lines of code, in addition to the insert\_before\_larger and sort\_array functions. All you have to do is:

* Call the sort\_array function to get a sorted version of A.
* Copy the values of the sorted result back to A.

The complete program should produce this output:

Before: [40, 10, 50, 20, 30]

After: [10, 20, 30, 40, 50]

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/assignment8/task7.java) contains an incomplete program. The goal of the program is to:

1. Start with an array list containing five strings.
2. Find the position of the shortest string in the array list, and remove that string.
3. Continue repeating step 2, until the array list is empty.

Complete that program, by defining two functions. First, define a find\_minimum\_length function, that satisfies the following specs:

* Function find\_minimum\_length takes one argument, called A, that is an array list of strings.
* The function should return the position of the shortest string in A.
* If A if empty, the function should return -1.

Second, define a remove\_minimum\_length function, that satisfies the following specs:

* Function remove\_minimum\_length takes one argument, called A, that is an array list of strings.
* The function should remove from A the shortest string in A.
* If A if empty, the function should do nothing.

Hints: my solution for the second function is four lines of code, on top of my solution for the first function. My solution for function remove\_minimum\_length calls function find\_minimum\_length.

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

The complete program should produce this output:

[whale, cat, elephant, donkey, goat]

minimum position = 1

[whale, elephant, donkey, goat]

minimum position = 3

[whale, elephant, donkey]

minimum position = 0

[elephant, donkey]

minimum position = 1

[elephant]

minimum position = 0

[]

minimum position = -1

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/assignment8/task8.java) contains an incomplete program. The goal of the program is to:

1. Create an empty array list of strings.
2. Insert into that array list some strings, one by one.
3. Insertions should be done in such a way that, after each insertion, the strings in the array list are sorted in ascending order of their length. In other words, shorter strings should appear before longer strings.

Complete that program, by defining an insert\_before\_longer function, that satisfies the following specs:

* Function insert\_before\_longer takes two arguments, called A, x. Argument A is an array list of strings, and argument x is a string.
* The function should look for the first element of A that is longer in length than x, and insert x right before that element.
* For example, if A contains elements "cat", "whale", "elephant", in that order, and x = "donkey", then x should be inserted at position 2, right before element "elephant". After the insertion, A should contain elements "cat", "whale", "donkey", "elephant", in that order.

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

Hint: this problem is very similar to that of Task 4, and my solution follows exactly the same steps as my solution for that task.

The complete program should produce this output:

[whale]

[cat, whale]

[cat, whale, elephant]

[cat, whale, donkey, elephant]

[cat, goat, whale, donkey, elephant]

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/assignment8/task9.java) contains an incomplete program. The goal of the program is to sort an array list of strings, so that the strings appear in ascending order of their length. Complete that program, by defining a sort\_by\_length function, that satisfies the following specs:

* Function sort\_by\_length takes one argument, called A, that is an array list of strings.
* The function should return an array list called result, with length equal to the length of A. The result should contain the same elements as A, but in ascending order of their length.

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

Hints: my solution also includes (and uses) the insert\_before\_longer function from the previous task. My solution takes five lines of code, in addition to the insert\_before\_longer function. Also, my solution is very similar to that of Task 5, and follows exactly the same steps as my solution for that task.

The complete program should produce this output:

Input: [whale, cat, elephant, donkey, goat]

Output: [cat, goat, whale, donkey, elephant]

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/assignment8/task10.java) contains an incomplete program. The goal of the program is to sort an array list of strings, so that the strings appear in ascending order of their length. Complete that program, by defining a sort\_by\_length2 function, that satisfies the following specs:

* Function sort\_by\_length2 takes one argument, called A, that is an array list of strings.
* The function MODIFIES A, so that it contains the same elements as before, but in ascending order of their length.

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

Hints: my solution also includes (and uses) the insert\_before\_longer and sort\_by\_length functions from the previous tasks. My solution takes four lines of code, in addition to the insert\_before\_longer and sort\_by\_length functions. Also, my solution is very similar to that of Task 6, and follows exactly the same steps as my solution for that task.

The complete program should produce this output:

Before: [whale, cat, elephant, donkey, goat]

After: [cat, goat, whale, donkey, elephant]

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 assignment8.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 assignment8.zip, follow these steps:

1. Create a folder called assignment8.
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 8 files: your answers document and all the Java code files (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 assignment8.zip?