**[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 2**

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 (5 pts.)**

public class task1 {

public static void main(String[] args) {

double x = 12.7;

x--;

int y = (int) Math.round(x);

y -= 5;

y++;

System.out.printf("y = %d\n", y);

}

}

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

**Task 2 (5 pts.)**

public class task2 {

public static void main(String[] args) {

double x = 12.7743;

x--;

x += 2;

System.out.printf("x = %.2f\n", x);

}

}

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

**Task 3 (10 pts.)**

public class task3 {

public static void main(String[] args) {

String a = "hello";

a += "!";

a += "123";

int b = a.length();

System.out.printf("a = %s\n", a);

System.out.printf("The length of a is %d characters.\n", b);

System.out.printf("The length of a is %d characters.\n", a.length());

}

}

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

**Task 4 (20 pts.)**

In a file called task4.java, write a program that:

* Asks the user to enter a double number (i.e., a floating-point number).
* Stores that number into a variable called a.
* Casts variable a into an integer, and stores the result into a variable called a1.
* Creates a variable a2, and sets it equal to the integer closest to a.
* Creates a variable a3, and sets it equal to the floor of a.
* Creates a variable a4, and sets it equal to the ceiling of a.
* Prints out the values of a1, a2, a3, a4, using EXACTLY the same format as shown below.

For example: if the user enters 12.8, your program output should look EXACTLY like this:

Please enter a double number: 12.8

a casted into an int becomes 12.

a rounded becomes 13.

The floor of a is 12.

The ceiling of a is 13.

As another example: if the user enters -7.8, your program output should look EXACTLY like this:

Please enter a double number: -7.8

a casted into an int becomes -7.

a rounded becomes -8.

The floor of a is -8.

The ceiling of a is -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 5 (10 pts.)**

import java.util.Scanner;

public class task5 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.printf("Please enter a double number: ");

double a = in.nextDouble();

int a1 = a;

System.out.printf("a casted into an int becomes %d.\n", a1);

}

}

The above program is an incorrect attempt to do the following:

* Ask the user to enter a double number (i.e., a floating-point number).
* Store that number into a variable called a.
* Cast variable a into an integer, and store the result into a variable called a1.
* Print out the value of a1.

Fix the above program, so that it runs correctly. Save your corrected program as task5.java. You do NOT have to explain what is wrong, it is sufficient to submit your corrected task5.java.

**Task 6 (10 pts.)**

In a file called task6.java, write a program that:

* Asks the user to enter a string that contains at least five letters.
* Stores that number into a variable called s.
* Creates a variable called "first", and sets it equal to the first letter of s.
* Creates a variable called "second", and sets it equal to the second letter of s.
* Creates a variable called "next", and sets it equal to the string that consists of the third, fourth, and fifth letter of s.
* Prints out the values of variables first, second, next, using EXACTLY the same format as shown below.

For example: if the user enters the string "DeQ81", your program output should look EXACTLY like this:

Please enter a string, at least 5 letters long: DeQ81

The first letter of s is D.

The second letter of s is e.

The third, fourth, and fifth letters are Q81.

As another example: if the user enters the string "Saturday", your program output should look EXACTLY like this:

Please enter a string, at least 5 letters long: Saturday

The first letter of s is S.

The second letter of s is a.

The third, fourth, and fifth letters are tur.

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 (20 pts.)**

In a file called task7.java, write a program that:

* Asks the user to enter a string that contains at least three letters.
* Stores that number into a variable called s.
* Creates a variable called "ending", and sets it equal to the string that contains the last three letters of s. Here you should think carefully how to do this, you may need more than one line of code to achieve this type (hint: to figure out where the last three letters are, you need to know the length of the string).
* Prints out the value of ending, using EXACTLY the same format as shown below.

For example: if the user enters the string "Saturday", your program output should look EXACTLY like this:

Please enter a string, at least 3 letters long: Saturday

The last three letters are day.

As another example: if the user enters the string "July", your program output should look EXACTLY like this:

Please enter a string, at least 3 letters long: July

The last three letters are uly.

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 (20 pts.)**

In a file called task8.java, write a program that:

* Asks the user to enter a string.
* Asks the user to enter a second string.
* Prints out the length of the first string, the length of the second string, and the sum of the two lengths, using EXACTLY the same format as shown below.

For example: if the user enters strings "UT" and "Arlington", your program output should look EXACTLY like this:

Please enter a string: UT

Please enter a second string: Arlington

The first string has length 2.

The second string has length 9.

The sum of the two lengths is 11.

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 assignment2.zip (no other forms of compression accepted, contact the instructor or TA if you do not know how to produce .zip files). The zipped directory should contain 6 files: your answers document and all the Java code files (task4.java, task5.java, task6.java, task7.java, task8.java).

To create a zipped directory called assignment2.zip, follow these steps:

1. Create a folder called assignment2.
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.

This sample [assignment2.zip](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment2/assignment2.zip) file shows how your assignment2.zip file should look like. Your zip file should have exactly the same structure as the sample file.

**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 assignment2.zip?