**[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 3**

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 main(String[] args)

{

int x = 4;

int y = 10;

boolean b1 = (x == y);

boolean b2 = (x == y) && (y > 8);

boolean b3 = (x == y) || (y > 8);

boolean b4 = !((x == y) || (y > 8));

boolean b5 = !b1;

boolean b6 = b1 && b2;

boolean b7 = b1 || b2;

boolean b8 = !(b1 && b2);

boolean b9 = false && (false || true);

boolean b10 = (false && false) || true;

System.out.printf("b1 = %b\n", b1);

System.out.printf("b2 = %b\n", b2);

System.out.printf("b3 = %b\n", b3);

System.out.printf("b4 = %b\n", b4);

System.out.printf("b5 = %b\n", b5);

System.out.printf("b6 = %b\n", b6);

System.out.printf("b7 = %b\n", b7);

System.out.printf("b8 = %b\n", b8);

System.out.printf("b9 = %b\n", b9);

System.out.printf("b10 = %b\n", b10);

}

}

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)

{

String s1 = "strawberry";

String s2 = s1.substring(0,5);

String s3 = "Straw";

boolean b1 = ("straw" == s2);

boolean b2 = "straw".equals(s2);

boolean b3 = ("straw".compareTo(s2) < 0);

boolean b4 = ("straw".compareTo(s2) == 0);

boolean b5 = ("straw".compareTo(s3) < 0);

System.out.printf("b1 = %b\n", b1);

System.out.printf("b2 = %b\n", b2);

System.out.printf("b3 = %b\n", b3);

System.out.printf("b4 = %b\n", b4);

System.out.printf("b5 = %b\n", b5);

}

}

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

**Task 3 (5 pts.)**

public class task3

{

public static void main(String[] args)

{

String s1 = "hello";

if (s1.equals("goodbye"));

{

System.out.printf("checkpoint 1\n");

}

if (s1.equals("goodbye"))

{

System.out.printf("checkpoint 2\n");

}

if (s1.equals("hello"))

{

System.out.printf("checkpoint 3\n");

}

else if (!(s1.equals("goodbye")))

{

System.out.printf("checkpoint 4\n");

}

else if (s1.equals("hello"))

{

System.out.printf("checkpoint 5\n");

}

}

}

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

**Task 4 (10 pts.)**

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

* Asks the user to enter two nouns and a verb, and saves them as variables called first\_noun, second\_noun, and verb.
* Prints a sentence that looks like this: The first\_noun verb over the second\_noun.

For example: if the user enters nouns "cat" and "table" and verb "jumps" your program output should look EXACTLY like this:

Enter the first noun: cat

Enter the second noun: table

Enter a verb: jumps

The cat jumps over the table.

As another example: if the user enters nouns "blah" and "qqq" and verb "bbb" your program output should look EXACTLY like this:

Enter the first noun: blah

Enter the second noun: qqq

Enter a verb: bbb

The blah bbb over the qqq.

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.)**

In a file called task5.java, write a program that:

* Asks the user to enter two real numbers (doubles, not integers) called N1 and N2.
* Computes and displays the following operations between the two: multiplication, division, exponentiation. For the results of these two operations, only two decimal digits should be displayed.

For example: if the user enters numbers 11 and 7, your program output should look EXACTLY like this:

Please enter real number N1: 11

Please enter real number N2: 7

11.000000 \* 7.000000 = 77.00

11.000000 / 7.000000 = 1.57

11.000000 raised to the power of 7.000000 = 19487171.00

As another example: if the user enters numbers 3.25 and 10.3, your program output should look EXACTLY like this:

Please enter real number N1: 3.25

Please enter real number N2: 10.3

3.250000 \* 10.300000 = 33.48

3.250000 / 10.300000 = 0.32

3.250000 raised to the power of 10.300000 = 187239.99

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.)**

A day has 86,400 secs (24 hrs \* 60 mins \* 60 secs). In a file called task6.java, write a program that:

* Prompts the user for an integer, specifying a number of seconds in the range 1 to 86,400.
* Outputs the time as hours, minutes, and seconds with a 24-hour clock.
* If the user inputs an integer that is less than 1 or greater than 86,400, your program should print "Invalid number of seconds, must be between 1 and 86400."

You can divide the user's value by 3600 (the number of seconds in an hour) to determine the number of hours, and then divide the remainder of that operation by 60 (the number of seconds in a minute) to calculate the number of minutes. The remainder of that operation will be the number of seconds.

For example, this is how your program would work step by step, if the user enters 70000:

* Prompt the user for a value in the range 1 to 86,400.
* The user inputs 70000 for this example.
* Divide the user input by 3600 to get 19 (hrs)
* Get the remainder of the above operation: there are 1600 seconds left over.
* Divide that remainder by 60 to get 26 (mins)
* Get the remainder of the above operation: there are 40 seconds left over.
* Therefore, we have 19 hours, 26 minutes, and 40 seconds.

For example: if the user enters 70000, your program output should look EXACTLY like this:

Please specify the number of seconds (between 1 and 86400): 70000

70000 seconds correspond to 19 hours, 26 minutes, and 40 seconds.

As another example: if the user enters 2345, your program output should look EXACTLY like this:

Please specify the number of seconds (between 1 and 86400): 2345

2345 seconds correspond to 0 hours, 39 minutes, and 5 seconds.

As another example: if the user enters 100000, your program output should look EXACTLY like this:

Please specify the number of seconds (between 1 and 86400): 100000

Invalid number of seconds, must be between 1 and 86400.

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.)**

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

* Asks the user to specify a month as an integer between 1 and 12.
* Prints out the name of the corresponding month.
* Prints out "invalid month" if the integer is less than 1 or greater than 12.

For example: if the user enters 1, your program output should look EXACTLY like this:

Please indicate a month as an integer from 1 to 12: 1

January

As another example: if the user enters 8, your program output should look EXACTLY like this:

Please indicate a month as an integer from 1 to 12: 8

August

As another example: if the user enters 54, your program output should look EXACTLY like this:

Please indicate a month as an integer from 1 to 12: 54

invalid month

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.)**

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

* Asks the user to specify the name of a month. This name should be either all in lower case letters, or with the first letter in upper case and all other letters in lower case.
* Prints out the order of that month in the year (e.g., January is the first month, July is the seventh month, and so on).
* Prints out "invalid month" if the user enters a name that is not recognized.

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

Please enter the name of a month: june

June is the sixth month.

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

Please enter the name of a month: May

May is the fifth month.

As another example: if the user enters "JUne", which violates our rules by starting with two uppercase letters, your program output should look EXACTLY like this:

Please enter the name of a month: JUne

invalid month

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.)**

In a file called task9.java, write a program that:

* Asks the user to enter an integer.
* If the integer is less than zero, the program prints that the number is negative.
* If the integer is greater than or equal to 0 and less than 1 million, the program prints how many digits the number has.
* if the number is greater than or equal to 1 million, the program prints that the number has more than six digits.

For example: if the user enters -7, your program output should look EXACTLY like this:

Please enter an integer: -7

-7 is negative.

As another example: if the user enters 0, your program output should look EXACTLY like this:

Please enter an integer: 0

0 has one digit.

As another example: if the user enters 8712, your program output should look EXACTLY like this:

Please enter an integer: 8712

8712 has four digits.

As another example: if the user enters 15151515, your program output should look EXACTLY like this:

Please enter an integer: 15151515

15151515 has more than six digits.

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.)**

In a file called task10.java, write a program that:

* Asks the user to enter an integer.
* If the integer is divisible by 2 and by 3, the program prints: "the number is even and divisible by 3."
* If the integer is divisible by 2 but not by 3, the program prints "the number is even and not divisible by 3."
* If the integer is divisible by 3 but not by 2, the program prints "the number is odd and divisible by 3."
* If the integer is divisible by neither 2 nor 3, the program prints "the number is odd and not divisible by 3."

For example: if the user enters 123, your program output should look EXACTLY like this:

Please enter an integer: 123

The number is odd and divisible by 3.

As another example: if the user enters 40, your program output should look EXACTLY like this:

Please enter an integer: 40

The number is even and not divisible by 3.

As another example: if the user enters 41, your program output should look EXACTLY like this:

Please enter an integer: 41

The number is odd and not divisible by 3.

As another example: if the user enters 42, your program output should look EXACTLY like this:

Please enter an integer: 42

The number is even and divisible by 3.

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 11 (10 pts.)**

In a file called task11.java, write a program that:

* Asks the user to enter a word.
* If word starts with a vowel, the program prints that it starts with a vowel.
* If word starts with a consonant, the program prints that it starts with a consonant.
* If word starts with neither a vowel nor a consonant (e.g., it starts with a number, or a punctuation character), the program prints that it starts with neither a vowel nor a consonant.

For example: if the user enters "543", your program output should look EXACTLY like this:

Please enter a word: 543

543 starts with neither a vowel nor a consonant.

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

Please enter a word: Airplane

Airplane starts with a vowel.

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

Please enter a word: cat

cat starts with a consonant.

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 assignment3.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 assignment3.zip, follow these steps:

1. Create a folder called assignment3.
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 [assignment3.zip](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment3/assignment3.zip) file shows how your assignment3.zip file should look like. Your zip file should have exactly the same structure as the sample file, so it should contain only 9 files: your answers document and all the Java code files (task4.java, task5.java, task6.java, task7.java, task8.java, task9.java, task10.java, task11.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 assignment3.zip?