**[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 5**

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 foo(int a, int b)

{

System.out.printf("1: a = %d\n", a);

System.out.printf("2: b = %d\n", b);

a = a\*a;

b = 2\*b;

System.out.printf("3: a = %d\n", a);

System.out.printf("4: b = %d\n", b);

}

public static void main(String[] args)

{

int a = 3;

int b = 5;

foo(b, a);

System.out.printf("5: a = %d\n", a);

System.out.printf("6: b = %d\n", b);

}

}

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

**Task 2 (10 pts.)**

public class task2

{

public static int foo(int a)

{

a = a+1;

return a;

}

public static void main(String[] args)

{

int a = 1;

int b = 5;

int c = foo(foo(b));

int d = foo(foo(foo(b))) + foo(foo(a)) + 10;

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

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

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

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

}

}

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 String foo(String a)

{

int m = a.length();

String result = a.substring(0, m/2);

return result;

}

public static void main(String[] args)

{

String b = "Arlington";

String c = foo(b);

String d = foo(foo(b));

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

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

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

}

}

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

**Task 4 (10 pts.)**

import java.util.Scanner;

public class task4

{

public static int count\_bs(String word)

{

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

{

int counter = 0;

if ((word.charAt(i) == 'b') || (word.charAt(i) == 'B'))

{

counter++;

}

}

return counter;

}

public static void main(String[] args)

{

Scanner in = new Scanner(System.in);

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

String str = in.next();

int result = count\_bs(str);

System.out.printf("%s contains letter B %d times.\n", str, result);

}

}

The program above is an INCORRECT attempt to write a program that:

* Asks the user to enter a word.
* Prints out the number of occurrences of the letter B (both upper and lower case) in that word.

In a file called task4.java, modify the above program so that it works correctly. **IMPORTANT: you are only allowed to modify the count\_bs function. You are NOT allowed to modify in any way the main function**.

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

Please enter a word: Babylon

Babylon contains letter B 2 times.

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

Please enter a word: airplane

airplane contains letter B 0 times.

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 (15 pts.)**

File [task5.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment5/task5.java) contains an incomplete program. Complete that program, by defining a sphere\_volume function, that satisfies the following specs:

* sphere\_volume takes one argument, called radius.
* The function returns the volume of a sphere with the specified radius, using the formula:
* volume = (4/3) \* π \* radius3

**IMPORTANT: you are only allowed to modify the provided code by writing the sphere\_volume function. You are NOT allowed to modify in any way the main function.**

This is an example run of the complete program:

Please enter a radius, or -1 to quit: 2

Volume = 33.51.

Please enter a radius, or -1 to quit: 6.1

Volume = 950.78.

Please enter a radius, or -1 to quit: -1

Exiting...

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 (15 pts.)**

File [task6.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment5/task6.java) contains an incomplete program. Complete that program, by defining an is\_leap\_year function, that satisfies the following specs:

* is\_leap\_year takes one argument, called year.
* if year is a leap year, then is\_leap\_year (year) returns true.
* if year is NOT a leap year, then is\_leap\_year (year) returns false.

**IMPORTANT: you are only allowed to modify the provided code by writing the is\_leap\_year function. You are NOT allowed to modify in any way the main function.**

Hint: our [LeapYear.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/codebase/05a_conditionals/LeapYear.java) program has code that you may find useful, and you should feel free to reuse.

This is an example run of the complete program:

Please enter a year, or -1 to quit: 1992

Yes, 1992 is a leap year.

Please enter a year, or -1 to quit: 2000

Yes, 2000 is a leap year.

Please enter a year, or -1 to quit: 1900

No, 1900 is not a leap year.

Please enter a year, or -1 to quit: 2015

No, 2015 is not a leap year.

Please enter a year, or -1 to quit: -1

Exiting...

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 (15 pts.)**

File [task7.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment5/task7.java) contains an incomplete program. Complete that program, by defining a pick\_middle function that satisfies the following specs:

* pick\_middle takes three arguments, called first, second, third.
* pick\_middle(first, second, third) returns the middle of the values of the three arguments.

**IMPORTANT: you are only allowed to modify the provided code by writing the pick\_middle function. You are NOT allowed to modify in any way the main function.**

This is an example run of the complete program:

please enter the first number: 5

please enter the second number: 5

please enter the third number: 5

the middle value is 5.0

please enter the first number: 5.1

please enter the second number: 3.2

please enter the third number: 7.5

the middle value is 5.1

please enter the first number: 7

please enter the second number: 8

please enter the third number: 9

the middle value is 8.0

please enter the first number: 3

please enter the second number: 3

please enter the third number: 4

the middle value is 3.0

please enter the first number: 4

please enter the second number: 4

please enter the third number: 3

the middle value is 4.0

please enter the first number: 4

please enter the second number: 3

please enter the third number: 3

the middle value is 3.0

please enter the first number: 1

please enter the second number: 88

please enter the third number: 15

the middle value is 15.0

please enter the first number: -4

please enter the second number: -1

please enter the third number: 56

the middle value is -1.0

please enter the first number: q

Exiting...

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 (15 pts.)**

File [task8.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/assignments/assignment5/task8.java) contains an incomplete program. Complete that program, by defining a starts\_with\_vowel function, that satisfies the following specs:

* starts\_with\_vowel takes one argument, called word.
* if word starts with a vowel, then starts\_with\_vowel(word) returns true.
* otherwise, starts\_with\_vowel(word) returns false.

**IMPORTANT: you are only allowed to modify the provided code by writing the starts\_with\_vowel function. You are NOT allowed to modify in any way the main function.**

Hint: our [StartsWithVowel2.java](http://vlm1.uta.edu/%7Eathitsos/courses/cse1310_fall2015/codebase/05a_conditionals/StartsWithVowel2.java) program has code that you may find useful, and you should feel free to reuse.

This is an example run of the complete program:

Enter some word, or q to quit: airplane

Yes, airplane starts with a vowel.

Enter some word, or q to quit: train

No, train does not start with a vowel.

Enter some word, or q to quit: Car

No, Car does not start with a vowel.

Enter some word, or q to quit: Elephant

Yes, Elephant starts with a vowel.

Enter some word, or q to quit: q

Exiting...

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 assignment5.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 assignment5.zip, follow these steps:

1. Create a folder called assignment5.
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 6 files: your answers document and all the Java code files (task4.java, task5.java, task6.java, task7.java, task8.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 assignment5.zip?