**IT 1090C Computer Programming I**

**Prof. Tom Wulf**

# Lab 4 Introduction to Java

**12/2 pts (2 extra credit pts)**   
  
Note: Complete the lab individually. You can get help from your group members but it is very important that each student complete the lab themselves using the JetBrains IntelliJ IDEA IDE. You want to be totally comfortable with creating java projects in IntelliJ.

# Learning Goals:

* how java’s numeric types, (int, and double), interact with the simple arithmetic operators (multiplication, addition, modulo, etc.).
* code several of our previous pseudocode programs in java!
* become comfortable with using IntelliJ IDEA IDE for our java programming
* get used to submitting assignments in Canvas
* optionally (if your instructor indicates it) use GitHub source control for your java projects from within IntelliJ.

# Part I: IntelliJ Project and Java Main File (2 pts)

1. Create a new IntelliJ java project called **Lab\_04\_Java\_Variables** use the default java 1.8 and don’t select anything else on the initial screen. *Note: your instructor will demonstrate this in class and there is a Video that demos it.*   
     
   (*We are not allowed to use spaces in project names for GitHub so we will get in the habit of avoiding them by using underscores \_.)*
2. Choose the Create Project from Template and the Command Line App template which will create a Main.java file. All your code will go in the main.java file
3. If your instructor indicated that you should use GitHub, add your project to GitHub
   1. Goto the VCS (Version Control System) on the main menu bar in IntelliJ and choose “Share Project on GitHub”
   2. Add a description usually “Project init” to indicate that you are just creating the project and click “share”. *(For this first time, I recommend that you open your GitHub account in the Web browser and confirm that your project was created and later, after you work on it, updated!!!)*

**Take a break and check with your fellow learners. Was everyone able to create the project files correctly? (If you are taking the course online, be sure to watch the support videos before you do this lab! Refer to them if you have trouble. Don’t delay with this. You can’t proceed with the course if you can’t use the IDE.)**

Now continue:

Click the arrow next to the **src** folder in your project to expand it. Now you can see the **main.java** file that was created from the template. All your code goes in this file within the method called main.  
  
Within your file declare and initialize a series of numeric variables. Put them one per line like this example:

*int intOperandA = 1:*

*… next variable declaration here…*

Here are the int (integer) variables to declare:

***intOperandA, intOperandB, intSum, intProduct, intDifference, intQuotient, intModulo  
  
You can set these to simple integer values of your own choice. Make them all different: 1 5, 13, 24, etc.***

**Next:**

In the same way, create a series of double value variables=s. (These can have fractional values like 3.50.)

*double doubleOperandA = 3.50; // This is a declaration and initial assignment*  
  
Create declarations again, one per line, for each of these.  
  
**doubleOperandA, doubleOperandB, doubleSum, doubleProduct, doubleDifference, doubleQuotient**

*Note that we do not include the type of variables i.e. int, double, boolean, etc in the name of the variables. I’m having you do that here this one time so you can observe how the typing effects the arithmetic operations.*

Now create assignment statements using your variables for each of these arithmetic operations for both the int and the double types:

+, -, \*, /, % (that last one is remainder and is called modulo) Create a ‘complete sentence’ output statement that displays the results of the operation and the two operands.

**Example:**

intSum = intOperandA + intOperandB; // Assignment that uses the arithmentic operator +

System.out.println(“The sum using ints of “ + intOperandA + “ “ + intOperandB + “ is “ + intSum); // Output statement

Do this with the int variables for each of the operations (+ sum, \* product, - difference, / quotient, and % modulo. **Notice how we include a space before and after each operator to increase readability.**

**Get in the habit of typing the code this way.**

**Get in the habit of typing the code this way.**

**Get in the habit of typing the code this way.**

Now do the same thing with the **double variables**

Now, execute (that is ‘run’) your program. Right click on your **main.java** file in the project view and choose **Run main**. Alternatively, you can run the program by clicking run on the top menu bar.

Size the output window so you can see the results of the **println** statements for these expressions and grab a screen shot of your desktop that shows this. Alternatively, and what often works better for me, copy the output window as text and paste it into this document.

[Windows: Alt + printScrn key puts image of desktop in clipboard just use past to insert it here in to this document.

Mac: I think Apple /Command + 2 or 3 does the same thing.]

**Be sure when you do these screen shots or output copies that they clearly show me the work I asked you to do. We use this to grade your work. It is assumed that you will completely test and debug your code before submitting it and the grading process is primarily simply confirming you completed the assigned work.   
  
If you have a problem completing the assignment, seek help from your instructor, the tutor, etc. Don’t wait for help, you will fall behind…**

**Insert the screen shot here:**

Did any of these results surprise you? (If not, take another look at the result of int division)

Next Part:  
  
OK complete this part by creating declarations and initializing each of these variables. Be sure to type the variable appropriately as int, double, or String.  
Choose meaningful names of your own for your variables and use camel case:

**Example:  
the cost of my lunch:** double myLunchCost = 12.50;

**the number of kids in your family:**

**the answer to is it raining? :**

**the price of a gallon of gas :**

**your favorite number :**

**your shoe size:**

**your birth month:**

**your full name:**

(If you want, you can print these out with java statements but I’ll make that optional so you can finish the lab during the session.)  
 **double myLunchCost = 12.50;  
System.out.println(“The cost of my lunch is: “ + myLunchCost);**

Get another screen shot or copy of your editor window making sure I can read all your variable declarations.

**Paste the screen image or output copy into this document here:**

If your instructor directed you to use GitHub:

*Click on GIT on the main menu bar and choose Commit. On the popup screen use the dropdown on the submit button to Commit and Push.*

Typically, you commit several times as you write more code but you Push at the end of your session to upload all the commits to the GitHub repository and back up your work. You can also use the Commits to restore your project to a previous point like the undo function in an editor.  
  
Paste the URL link for the GitHub repo here:

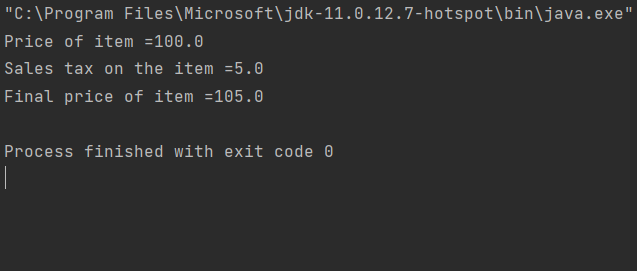
Part II:

Now code each of these programs. *Since we can’t do java input yet, just set the input value as a literal value in your code.* I’ll leave it to you to choose the values. Be sure to use clear and meaningful variable names with appropriate types! Try to pick values that make it easy for you to visually confirm the results of your program outputs.  
  
Here is an example:  
  
We did an example where the user enters a number and the program doubles it. Here is the code that I would put within the main() class of my IntelliJ project:  
  
**int myFavNum = 23; // Later we would input this but today we simulate that  
int doubleNum = 0;**

**doubleNum = myFavNum \* 2;  
  
System.out.println(“The number “ + myFavNum + “ doubled is: “ + doubleNum);**When you run the code, you will see the println statement with the variable names replaced with the actual values.

On with the lab:  
  
Create a **new** IntelliJ Java Project for **each** of these programs using the specified names.  
  
*If your instructor directed you to use GitHub, create a GitHub project for each program as with the previous one. Be sure to commit and push you project when completed. Again, this first time, I recommend that you access your GitHub account with a web browser and confirm that you projects are created an updated.*

**Task 1 (3 pts):   
Project name: Lab\_04\_01\_SalesTax**Given the price of a purchase the program computes and outputs a 5% sales tax. Be sure to use output statements that clearly display the price and the computed sales tax. (No conditions here, the sales tax is always the same rate.)

**  
  
Screenshots or output copy here from Task 1 (Be sure to include a readable shot of your programs output for each of these tasks):**

**Don’t forget to commit and push your project in GitHub if you were told to use it.  
Post the URL link for this repository here:  
  
  
Task 2 (3 pts):   
Project: Lab\_04\_02\_YearlyMaintenanceCost**

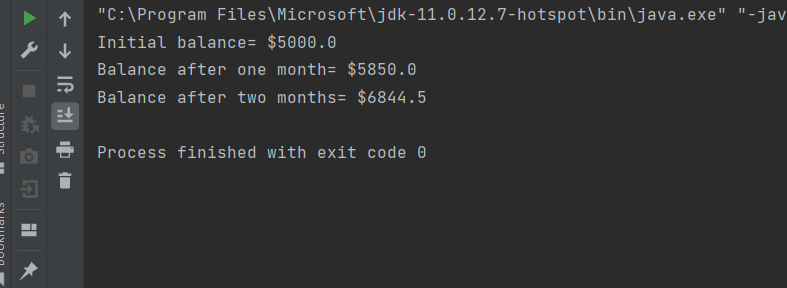
A program that uses home maintenance costs for each of the four seasons and returns the total yearly maintenance costs. Again, be clear in the output by echoing the values for the seasonal costs when you display the combined total.  
 **Screenshots or output copy here for Task 2:**

**Text

Description automatically generated**

**Don’t forget to commit and push your project in GitHub if you were told to use it. Post the URL link for this repository here:  
  
  
  
Task 3 (4 pts):   
Project: Lab\_04\_03\_MonthyCCBalance**A program that starts with a credit card balance of $5000 and then calculates the interest at a rate of 17%. Assume the user does not make any payments. Display the interest due after one month and again after two months.

**Screenshots or output copy here for Task 3:**

**  
Don’t forget to commit and push your project in GitHub if you were told to use it. Post the URL link for this repository here:**

**Task 4 (2 pts): (Extra Credit)   
Project: Lab\_04\_04\_EvenOdd**  
A program that takes a integer (java int) and uses the modulo function to indicate if the number is odd or even. Since we can’t do input in java yet, just hard code the value for the input as a variable instantiatio**n.**

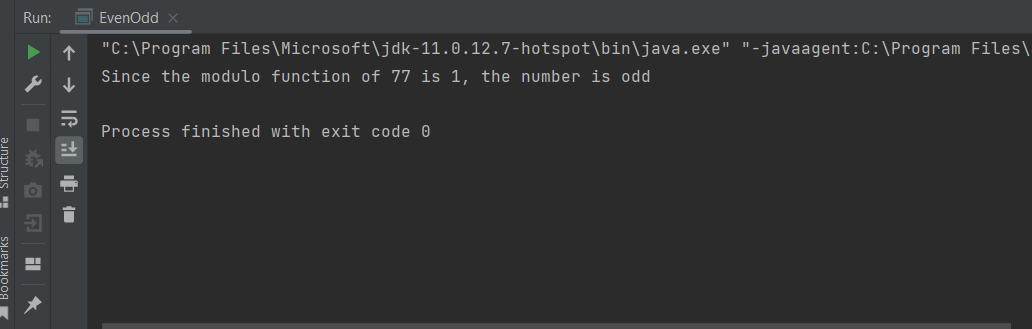
**(Instead of inputing the value which we can’t do yet, set the value in the code.**

**int numToExamine = 2;**

**// program will then determine that since 2 mod 2 is 0 that numToExamine is Even.**

**)**

**Screenshots or output copy here for Task 4:**

**  
Don’t forget to commit and push your project in GitHub if you were told to use it. Post the URL link for this repository here:**

Submitting your work: **PLEASE READ CAREFULLY. NOW THAT WE ARE USING JAVA YOU WILL HAVE TO SUBMIT ALL THE WORK.**

Make sure that you saved your screen shots or output copy in this document and that this file is correctly named: **Lastname\_Firstname\_Lab\_04.docx** using your name. Everyone submits this file every time in Canvas. We use it to grade your work and will not do so if you fail to submit it as directed.

If your instructor directed you to use GitHub:  
  
 Submit **each of** the GitHub project links for each of the 5 tasks in canvas. You might need to use a canvas comment for this. This is in addition to the MS Word docx which everyone submits every time.  
  
If you are not using GitHub:  
  
Create a new compressed \*.zip archive folder. (Don’t give me any other type of archive, it will be returned to you ungraded!) Name it: **Lastname\_Firstname\_Lab\_04.zip** using your name. Place a copy of each of your entire Netbeans project folder(s) and a copy of this doc file within your .zip archive. When you submit your lab, upload the complete archive as well as the **separate** copy of this MS Word file which will be used for grading your lab. **This is very important. If you submit the word file, we can insert grading feedback to you directly in it.**  
  
Although I won’t run your code, because it is your job to test and debug it, I ask that you give me your entire project every time so I can run your code or examine it if I need to for grading. **If you don’t follow the submission directions, your work will not be graded.**  
Use the Canvas Assignment mechanism to submit your docx file. Submit either the .zip archive with all the code or a text file with your GitHub repo links.  
  
For the Extra credit resubmit your entire assignment using the separate the extra credit submission link. This keeps the extra credit separate in the gradebook and also insures that I grade it.  
  
**Every assignment for the rest of the course will follow these submission guidelines. Please get it right!**