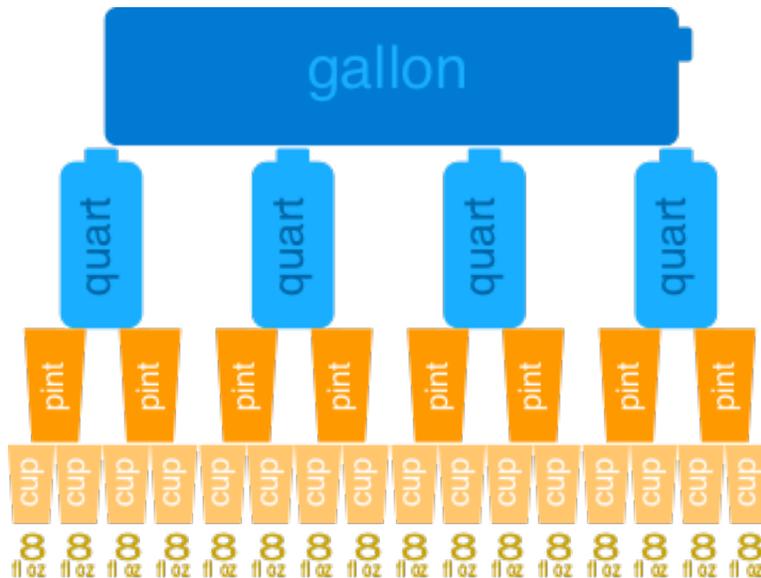


# Assignment 03

CS 120: Software Design I

## Program 1: VolumeConverter

Your first program will be written in a class called `VolumeConverter`. When measuring out the volume of liquids, we use a number of different units, such as gallons or ounces. These values are related to one another via various conversions. For example, one gallon is equal to four quarts. You can see a summary of the relationship between these different units below:



In essence, 1 gallon is equal to 4 quarts, 8 pints, 16 cups, and 128 fluid ounces. Your program will take in an `int` value of fluid ounces (the smallest unit), and then print out the equivalent value in the largest unit for which it has at least one of that unit (e.g., 9 fluid ounces would be printed as 1.12 cups because it can fulfill at least one whole value of that unit, and not 0.56 pints) to two decimal places. Your program should support conversion of fluid ounces to gallons, quarts, pints, cups, and fluid ounces.

Below are five example runs of the program with input **Highlighted**. Make sure that your output matches the examples below (except for the sections determined by user input).

### Example 1

```
Enter the number of fluid ounces to convert: 14
```

```
14 fluid ounces is equal to 1.75 cups
```

### Example 2

```
Enter the number of fluid ounces to convert: 345
```

```
345 fluid ounces is equal to 2.70 gallons
```

### Example 3

```
Enter the number of fluid ounces to convert: 96
```

```
96 fluid ounces is equal to 3.00 quarts
```

**Example 4**

```
Enter the number of fluid ounces to convert: 7
```

```
7 fluid ounces is equal to 7.00 fluid ounces
```

**Example 5**

```
Enter the number of fluid ounces to convert: 32
```

```
32 fluid ounces is equal to 1.00 quart
```

Some prompts and questions to consider *before* starting to write the program:

- Starting with the examples above, how do you convert from fluid ounces to the given unit by hand? Next, pick some additional input values and try to convert those by hand too. How do you perform the conversion? How do you decide which unit to convert to?
- Consider Examples 1 and 2, which output decimal values. Redo your conversions by hand, paying careful attention to the idea of integer mathematical operations as done in Java. Why do your answers differ from the examples above? How can you perform the conversions in your program to ensure you get the correct answer?
- Sometimes, your conversion will result in exactly one of some unit (e.g., Example 5). How can you output the singular value of that unit (e.g., quart) instead of the plural value (e.g., quarts)?
- How can you ensure exactly two decimal places are always printed?

The following is a high level checklist of requirements for your program:

- Your class is named `VolumeConverter` and is placed in the project created for this assignment
- Your code is commented according to the guidelines in the Java Style Guide found on D2L
- Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- Your code fulfills the functionality outlined above
- You have a blank line between the prompts and the output
- Your strings (both prompts and output) are formatted **exactly** as shown in the examples

## Program 2: Sort3Numbers

The second program you will be writing will be in a class named `Sort3Numbers`. Sorting values is a common operation, from sorting numbers in a spreadsheet to sorting grades for a class. Sorting two values is easy to do by hand; however, sorting becomes more difficult as the number of values increases. For this program, you will be soliciting three values from the user, and then printing them out in sorted order.

Your program will first prompt the user for their three values. The program will then print out the three values in sorted order.

Below are three example runs of the program with input **Highlighted**. Make sure that your output matches the examples below (except for the sections determined by user input).

### Example 1

```
Enter your first number: 1
Enter your second number: 2
Enter your third number: 3

Sorted numbers: 1, 2, 3
```

### Example 2

```
Enter your first number: 18
Enter your second number: 10
Enter your third number: 60

Sorted numbers: 10, 18, 60
```

### Example 3

```
Enter your first number: 9
Enter your second number: 9
Enter your third number: 3

Sorted numbers: 3, 9, 9
```

Some prompts and questions to consider *before* starting to write the program:

- How might the order in which values are entered affect the way you sort the values? For example, how can you ensure you get the same answer if the values are entered 3, 1, 2 or 3, 2, 1 or 1, 3, 2?
- What will happen if two of the values are the same? What about all three values?

The following is a high level checklist of requirements for your program:

- Your class is named `Sort3Numbers` and is placed in the project created for this assignment
- Your code is commented according to the guidelines in the Java Style Guide found on D2L
- Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- Your code fulfills the functionality outlined above
- You have all the prompts listed in the examples in the exact same order

- You have blank lines as indicated in the examples
- Your strings (both prompts and output) are formatted **exactly** as shown in the examples

### Program 3: LabPolicy

Your third program will be in a class named `LabPolicy`. Our CS120 class has a multi-faceted turn-in policy to accommodate where students are in the course; in this program, we'll be working with a **modified** version of the policy. Feel you have mastered the material and don't need to attend lab? Then turn in the third program. Want some extra help in understanding the material? Come to lab, get help, and turn in the first program. For this program, you will be implementing a program that informs users as to whether or not they successfully fulfilled the requirements for completing lab for the week<sup>1</sup>.

Your program will prompt the user for three values: whether or not they attended lab, whether or not they turned in the program (first program if they attended lab, third program if they did not), and whether or not the program was submitted on time (only prompt if they did submit a program). The program will then display output appropriate for the user's responses: if everything was done correctly, the program will display an affirmative message (the same message will be displayed for those who attended lab and submitted program 1 or did not attend and submitted program 3). If something was done incorrectly, the program will display a negative message and then offer a line of help specific to the reason for not completing the lab.

Below are six example runs of the program with input **Highlighted**. Make sure that your output matches the examples below (except for the sections determined by user input).

#### Example 1

```
Did you attend lab this week? [y / n] y
Did you submit program 1 to D2L? [y / n] y
Did you submit the program by the deadline? [y / n] y

Congratulations! You fulfilled your lab obligations for the week.
```

#### Example 2

```
Did you attend lab this week? [y / n] Y
Did you submit program 1 to D2L? [y / n] y
Did you submit the program by the deadline? [y / n] N

Sorry, you did not fulfill your lab obligations for the week.
Please try to get your program submitted on time in the future!
```

#### Example 3

```
Did you attend lab this week? [y / n] y
Did you submit program 1 to D2L? [y / n] n

Sorry, you did not fulfill your lab obligations for the week.
You need to submit your program in order to get credit.
```

<sup>1</sup>Whether they wrote their program correctly is not considered in this program. ©

**Example 4**

```
Did you attend lab this week? [y / n] n
Did you submit program 3 to D2L? [y / n] y
Did you submit the program by the deadline? [y / n] y

Congratulations! You fulfilled your lab obligations for the week.
```

**Example 5**

```
Did you attend lab this week? [y / n] N
Did you submit program 3 to D2L? [y / n] Y
Did you submit the program by the deadline? [y / n] N

Sorry, you did not fulfill your lab obligations for the week.
Please try to get your program submitted on time in the future!
```

**Example 6**

```
Did you attend lab this week? [y / n] n
Did you submit program 3 to D2L? [y / n] n

Sorry, you did not fulfill your lab obligations for the week.
Perhaps you should attend lab in the future.
```

Some prompts and questions to consider *before* starting to write the program:

- Draw out an activity diagram for this program. Walk through each of the different inputs given in the examples above. Do you arrive at the same outcomes as the examples?
- Looking at your activity diagram, consider how you might reuse states to avoid duplicating code. For example, there are several different sets of inputs that ultimately arrive at a negative message.
- How will you determine whether or not to display particular prompts (e.g., asking if the program was submitted by the deadline).
- How will you print the same negative message if the user did not complete the lab requirements for the week, but provide a different line of help depending on the situation?
- Consider how you might use variables to your advantage in this program, particularly in how you name your variables, such that your conditional statements read almost like English (e.g., “if the student did not attend lab and submitted program 3 and submitted the program on time...”).
- Note that in the examples, users can provide their answer with either an uppercase or lowercase letter. How might you accommodate both kinds of answers? You can check for both types of answers in your conditionals. A more elegant (but not required!) solution might employ `<String>.toUpperCase()` method to convert String values to uppercase (e.g., `“hello”.toUpperCase()` will return `“HELLO”`); there is also a `<String>.toLowerCase()` method for converting a String value to lowercase.

The following is a high level checklist of requirements for your program:

- Your class is named `LabPolicy` and is placed in the project created for this assignment
- Your code is commented according to the guidelines in the Java Style Guide found on D2L
- Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- Your code fulfills the functionality outlined above
- You have all the prompts listed in the examples in the exact same order
- You have blank lines as indicated in the examples
- Your strings (both prompts and output) are formatted **exactly** as shown in the examples