

**Due:** Monday, September 5, 2016 by 11:59 PM

## Deliverables

Your project files should be submitted to Web-CAT by the due date and time specified above (see the Lab Guidelines for information on submitting project files). In order to avoid a late penalty for the project, you must submit your completed code files to Web-CAT by 11:59 PM on the due date. You may submit your project up to 24 hours after the due date, but there is a late penalty of 15 points. No projects will be accepted after the one day late period. If you are unable to submit via Web-CAT, you should e-mail your project Java files in a zip file to your lab instructor before the deadline.

Files to submit to Web-CAT:

- WeightConversion.java
- FormulaCalculator.java

## Specifications

**Overview:** You will write two programs this week. The first will calculate the integral number of units (tons, pounds, and ounces) in a specified number of ounces, and the second will calculate the result of formula for inputs x, y, and z.

- **WeightConversion.java**

**Requirements:** A weight enthusiast would like a program that allows the user to enter a value in ounces and then displays the combination of tons, pounds, and ounces that each value is maximized in order by tons, pounds, and ounces. The input value should not exceed 500 million ounces (500,000,000).

**Design:** The weight enthusiast would like the output to look as shown in the examples below where 1234567890 is entered as the input for one run and 123456789 is entered for another run.

| Line number | Program output                        |
|-------------|---------------------------------------|
| 1           | Enter weight in ounces: 1234567890    |
| 2           | Limit of 500,000,000 ounces exceeded! |
| 3           |                                       |

| Line number | Program output                                      |
|-------------|---|
| 1           | Enter weight in ounces: 123456789                   |
| 2           | Combined Number of Tons, Pounds, and Ounces:        |
| 3           | Tons: 3858  |
| 4           | Pounds: 49  |
| 5           | Ounces: 5   |
| 6           | 123456789 ounces = 3858 tons + 49 pounds + 5 ounces |
| 7           |   |

Your program must follow the above format with respect to the output. Note that lines 3 through 6 for the input value of 123456789 begin with tab which is equivalent to three spaces in jGRASP (i.e., your output should use the **escape sequence for a tab**).

**Code:** In order to receive full credit for this assignment, you must read in a value and store it in a variable of type *int*, calculate the number of each unit (tons, pounds, ounces) and store each value in a variable of type *int*. It is recommended as a practice that you do not modify any input values once they are stored. Note that a ton contains 32,000 ounces and a pound contains 16 ounces. Your expressions should contain only *int* variables or *int* literals. Commas are not allowed in Java numeric literals (32000 may be entered as 32000 or 32\_000).

**Test:** You will be responsible for testing your program, and it is important to not rely only on the example above. The amount entered can be any valid *int* value. Consider testing with the input values such as 32000, 32016, 32017, etc.

- **FormulaCalculator.java**

**Requirements:** A program is needed that inputs values for x, y, and z and calculates a result for the indicated formula. If the product of x, y, and z is zero, the result is zero.

**Design:** The result should be calculated as follows:

$$result = \frac{(3x + 10.5) (5y - 6.2) (7z + 1.5)}{xyz} \quad \text{for } xyz \neq 0$$

Special case:

$$result = 0 \quad \text{for } xyz = 0$$

Three examples of program output for the indicated input values are show below. Note that lines 2 through 4 for the input values begin with tab which is equivalent to three spaces in jGRASP (i.e., your output should use the **escape sequence for a tab**)

**Example #1**

| Line number | Program output                                   |
|-------------|--|
| 1           | result = (3x + 10.5) (5y - 6.2) (7z + 1.5) / xyz |
| 2           | Enter x: 1                                       |
| 3           | Enter y: 1                                       |
| 4           | Enter z: 1                                       |
| 5           | result = -137.70000000000002                     |
| 6           |  |

**Example #2**

| Line number | Program output                                   |
|-------------|--|
| 1           | result = (3x + 10.5) (5y - 6.2) (7z + 1.5) / xyz |
| 2           | Enter x: 0                                       |
| 3           | Enter y: 1                                       |
| 4           | Enter z: 2                                       |
| 5           | result = 0.0                                     |
| 6           |  |

**Example #3**

| Line number | Program output                                   |
|-------------|--|
| 1           | result = (3x + 10.5) (5y - 6.2) (7z + 1.5) / xyz |
| 2           | Enter x: -1.5                                    |
| 3           | Enter y: 2.6                                     |
| 4           | Enter z: 3.7                                     |
| 5           | result = -77.47193347193347                      |
| 6           |  |

**Code:** Your numeric variables should be of type double. Use an if-else statement to determine if the divisor in the formula is zero. Note that in Example #2, the value of x is zero so the divisor evaluates to zero.

**Test:** You are responsible for testing your program, and it is important to not rely only on the examples above. Remember that the input values are doubles, so be sure to test both positive and negative values (with and without a decimal point) for x, y, and z. You should use a calculator or jGRASP interactions to check your answers.

**Grading**

**Web-CAT Submission:** You must submit both “completed” programs to Web-CAT at the same time. Prior to submitting, be sure that your programs are working correctly and that they have passed Checkstyle. **If you do not submit both programs at once, Web-CAT will not be able to compile and run its test files with your programs which means the submission will receive zero points for correctness.** I recommend that you create a jGRASP project and add the two files. Then you will be able to submit the project to Web-CAT. Activity 1 (pages 5 and 6) describes how to create a jGRASP project containing both of your files.