Academic Honesty Rules for This Assignment

This assignment is individual work. You may:

- Use the textbook and class videos, including code in the textbook and videos.
- Use the documentation at docs.oracle.com/en/java/ as a resource. Do not borrow code from the documentation.
- Ask me questions.

Discussing this assignment with any other person (in person or on the Internet), searching for answers on the Internet, or using sources other than the ones listed above is a violation of the class academic honesty policy. Please ask if you have any questions about this.

Specifications

Problem Description

You are going to write a program that computes the user's weight (in kg) on Mercury, Venus, and Mars, and the distance each planet will travel (in km) in a number of hours.

Input: You will obtain a weight (on Earth) in kilograms and the number of hours of the trip from the user. You may assume that the user will enter a positive double for each value.

Output: The program will display the entered weight converted to weight on Mercury, Venus, and Mars, and the weight on the other planet expressed as a percentage of the weight on Earth (e.g. weight on Mercury is approximately 38% weight on Earth). The program will display the distance each planet will travel, in kilometers, in the number of hours given.

Interface: This is a console program. Do not use JOptionPane. Your program should use the following prompts in the user interface:

```
Please enter your weight in kg on Earth:
Please enter the duration, in hours, of the trip:
```

The output should use the following text:

```
Your weight on Mercury is 16.989795918367346 kg, which is 37.755102040816325 percent
of your weight on Earth.
In 24.0 hours, Mercury will travel 4095360.0 km.
```

The underlined values are inputs, calculated values, or specific planets. The output should use the same text for each planet, substituting in the correct values and the correct planet names.

Design specifications: Use only techniques that are covered in the first lesson. Use console input and output. Do not, under any circumstances, use `System.exit`, `return`, or `break`. Do not submit code that includes these constructs. Submit whatever you can write that does not include these constructs. As we have not covered input validation techniques, no input validation is necessary. You may assume positive, floating-point (double) inputs.

You can use the data from this table from NASA. I've reproduced the relevant data here.

https://nssdc.gsfc.nasa.gov/planetary/factsheet/

	Mercury	Venus	Earth	Mars
Gravity (m/s^2)	3.7	8.9	9.8	3.7
Velocity (km/s)	47.4	35.0	29.8	24.1

Weight on a particular planet is calculated as mass * gravity.

Because the person will be entering their weight on Earth, you must divide that by 9.8 in order to determine their mass before multiplying by the gravity on a different planet.

For example, if something weighs 45 kg on Earth, we can calculate its weight on Mercury using this formula:

$$3.7 \times \frac{45}{9.8}$$

Resulting in a mere 16.9 kg on Mercury.

Note that $\frac{planet\ gravity}{Earth\ gravity} \times 100$ is the percent of an Earth weight that is the weight on another planet.

For the distance traveled in a number of hours, multiply the number of hours by the planet's velocity (given in km/s), and then by 60 (s/m), and then 60 (m/h) to obtain the number of kilometers traveled in the number of hours given.

Be careful to choose appropriate data types for your intermediate values and the results.

Because we have not yet covered output formatting, you should not truncate your numbers to a specific decimal place and your output will look messy.

Clarifications: Include in your submission everything you've clarified with the professor.

Assumptions: Include in your submission anything you assumed but didn't clarify. Include a justification.

Testing

Test Case 1

Purpose

Test the program on typical, realistic input values.

Input

Weight: 45
Hours: 24

Expected output

```
Your weight on Mercury is 16.989795918367346 kg, which is
37.755102040816325 percent of your weight on Earth.
Your weight on Venus is 40.86734693877551 kg, which is
90.81632653061224 percent of your weight on Earth.
Your weight on Mars is 16.989795918367346 kg, which is
37.755102040816325 percent of your weight on Earth.
In 24.0 hours, Mercury will travel 4095360.0 km.
In 24.0 hours, Venus will travel 3024000.0 km.
In 24.0 hours, Earth will travel 2574720.0 km.
In 24.0 hours, Mars will travel 2082240.0000000002 km.
```

Your program running

Paste your program executing in the testing document here. Show a complete execution: both inputs and output.

Conclusions

Describe whether your program's output matches expectations.

Test Case 2

Purpose

Test the program on minimum input values.

Input

Weight: 0

Hours: 0

Expected output

```
Your weight on Mercury is 0.0 kg, which is 37.755102040816325 percent
of your weight on Earth.
Your weight on Venus is 0.0 kg, which is 90.81632653061224 percent of
your weight on Earth.
Your weight on Mars is 0.0 kg, which is 37.755102040816325 percent of
your weight on Earth.
In 0.0 hours, Mercury will travel 0.0 km.
In 0.0 hours, Venus will travel 0.0 km.
In 0.0 hours, Earth will travel 0.0 km.
In 0.0 hours, Mars will travel 0.0 km.
```

Your program running

Paste your program executing in the testing document here. Show a complete execution: both inputs and output.

Conclusions

Describe whether your program's output matches expectations.

Test Case 3

Purpose

Test the program on the value 1 for each input.

Input

Weight: 1

Hours: 1

Expected output

```
Your weight on Mercury is 0.37755102040816324 kg, which is
37.755102040816325 percent of your weight on Earth.
Your weight on Venus is 0.9081632653061225 kg, which is
90.81632653061224 percent of your weight on Earth.
Your weight on Mars is 0.37755102040816324 kg, which is
37.755102040816325 percent of your weight on Earth.
In 1.0 hours, Mercury will travel 170640.0 km.
In 1.0 hours, Venus will travel 126000.0 km.
In 1.0 hours, Earth will travel 107280.0 km.
In 1.0 hours, Mars will travel 86760.0 km.
```

Your program running

Paste your program executing in the testing document here. Show a complete execution: both inputs and output.

Conclusions

Describe whether your program's output matches expectations.

Turn in

1. Your java program (the .java file only). Make sure that it meets the requirements of the rubric.

2. Complete the provided Assumptions and Testing document, which includes the following:

   Section 1: Clarifications and assumptions

   If you clarified specifications or made assumptions that were not clarified with the professor prior to submission, include them here with justifications.

   Section 2: Testing

   Copy / paste your program running on all test cases and include any additional test cases you used to thoroughly test your program. Answer any test case questions completely.

Zip (do not use any other compression format) the java file and the Assumptions and Testing document into a folder named JavaProgram1_your initials, for example, JavaProgram1_msb.

Upload the zipped folder to Blackboard.

Rubric

An exceptional-quality assignment will meet the following standards:

- Meeting functional and design specifications, 70 points
  The Java program works and meets all of the specifications, with no additional unspecified functionality. The programmer has used programming techniques from the first lesson only. If the program misses specifications or does not function correctly, errors are acknowledged with a thorough and reflective analysis in the testing section (points will be removed for missed specifications).
- Communicating with identifiers and white space, 10 points
  The program makes appropriate use of variables. Variables and constants are named according to convention and are named for understandability and purpose. White space, both vertical and horizontal, is correctly used for readability and meets programming conventions.
- Communicating through documentation, 10 points
  The Java program contains comments including the programmer's name and date. There are block comments (as many as necessary) for each distinct block of code which accurately describe what the block is accomplishing by relating the code to the problem being solved. Javadoc is included and meets the javadoc standards.
- Assumptions and Testing, 10 points
  Testing is thorough. If there are errors, they are described in the testing section. If there are questions, they are answered thoughtfully in the testing section. All assumptions are made explicit. The assignment will be reduced by a minimum of 10 points if the programmer claims a non-working program meets specifications.