# 07.03 Assignment Instructions

**Instructions:** Write a program to determine the surface gravity (g) on each planet in our solar system. (The results of this assignment will be used in later lessons, but will not be submitted for a grade at this time.)

1. Create a new project called 07 Defining New Methods in the Mod07 Assignments folder.
2. Create a class called PlanetGravity in the new project folder.
3. Read the Background section below to learn how to calculate each planet's surface gravity.
4. Look over the planet data below that contains the diameter and mass of each planet. Be sure to note the units of each value.
5. Assign the names of the planets to an array. Do the same for mass and diameter. Use a separate array for each.
6. Use static methods to break your program down into functional sections. For each method, determine whether parameters are needed or a return value.
7. Create a method to calculate the surface gravity for each planet. Depending on the accuracy of your information source, calculated values should be reasonably accurate but may vary somewhat. Be sure to double check your units.
8. Make a method to display the information in a neatly formatted table. (See sample output.)
9. Create a method to write the surface gravity data to a text file. When you inspect the file with a text editor, there should be eight numbers, one on each row. (You will need this file in the future.)

**Background:** Even if you have not taken physics, you have an intuitive understanding of gravity, the force that causes an object to fall when you drop it and keeps us from drifting off into space.

The surface gravity (g) of every planet is different, but can be easily calculated using two basic measurements: the planet's mass (M) and radius (r). If you have taken physics, the following equation will be very familiar. If not, simply look at it arithmetically; it **only involves** multiplication and division.

The following example shows how to calculate the Earth's surface gravity:

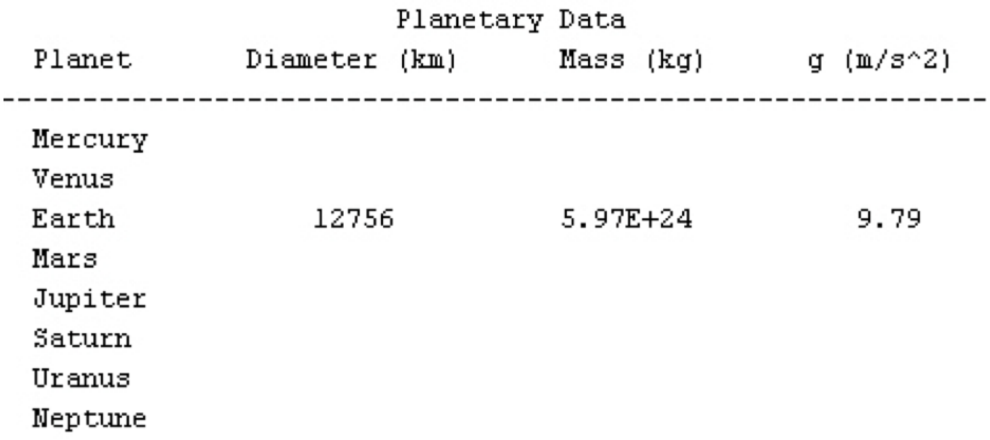
where g is the surface gravity in m/s$^2$,
G is the Universal Gravitational Constant,
M is the mass of the planet in **kilograms**,
r is the radius of the planet in **meters**.

$$
\begin{aligned}
g &= \frac{G*M}{r^2} \\
&= \frac{(6.67*10^{-11})*(5.98*10^{24})}{(6.378*10^{6})^2} \\
&= \frac{3.99*10^{14}}{4.07*10^{13}} \\
&= 9.81\ m/s^2
\end{aligned}
$$

**Planet Data:** In order to complete this assignment, fill in the missing data for the diameter and mass of each planet. If you locate information for the surface gravity of each planet, record it in the last column so you can compare the accuracy of your program's results.

| Planet | Diameter (km) | Mass (kg) |
|---|---|---|
| Mercury | 4,880 | 3.30e23 |
| Venus | 12,103.6 | 4.869e24 |
| Earth | 12,756.3 | 5.972e24 |
| Mars | 6,794 | 6.4219e23 |
| Jupiter | 142,984 | 1.900e27 |
| Saturn | 120,536 | 5.68e26 |
| Uranus | 51,118 | 8.683e25 |
| Neptune | 49,532 | 1.0247e26 |

**Expected Output:** When your program runs correctly, the output should resemble the following screen shot. Only data for Earth are shown here to help you with formatting, but each column of your output will be filled in.



```
                        Planetary Data
  Planet       Diameter (km)      Mass (kg)      g (m/s^2)
------------------------------------------------------------
  Mercury
  Venus
  Earth            12756          5.97E+24          9.79
  Mars
  Jupiter
  Saturn
  Uranus
  Neptune
```

* The carat symbol (**^**) is used in the heading of the last column to indicate that the unit for seconds (s) is squared. In some languages, arithmetic expressions can be raised to a power with the carat symbol, but in Java you must use the **`Math.pow()`** method. In this case, the **^** is part of a **String** literal, not an arithmetic expression.

** The data in column three are printed in scientific notation (e.g. 5.97E+24). The **`printf()`** method has additional type converters that can be used in the format specifier. The letter "E" (upper- or lowercase) is the conversion specifier for scientific notation (e.g. %7.2E).

Print