**Build Your Own Solar System!!!**

20 points: 10 for completing the worksheet as you go

 + 10 for implementing the code for the first planet

 + 5 Extra Credit for the second planet

You are going to write a new method to populate your scenario with the Sun and your chosen Planet!

The name of your method should be: public void sunAndPLANET() where “PLANET” should be the name of your chosen planet.

In order to do this, your method will add two bodies. Each will have its own:

* Size
* Mass
* Vector (which has two parts: speed and direction)
* X coordinate
* Y coordinate

You will also need to adjust the GRAVITY constant.

Just follow the steps below to create your new solar system!

**Preparatory Work:** After choosing a planet, find the following information:

* Mass of the Sun
* Mass of your planet
* Distance of your planet from the Sun
* Length of one year for your planet. That is, how long it takes your planet to orbit the Sun once.

One excellent source for this information is the following NASA site: http://solarsystem.nasa.gov/planets/index.cfm

Remember to use consistent units!!!!

**First Step:** Make a copy of your working Newton's Lab project and call it LastNameFirstNamePLANET where “PLANET” is the name of your chosen planet.

Now for your Sun and Planet Body:

**Parameters 1 & 4: SIZE and COLOR**

**SIZE:** For the size numbers in your scenario, just copy the sizes from the sunAndPlanet() method. This is just the size of the picture in the scenario, and does not represent any actual physical properties.

(If you used actual proportions, you would not be able to see your planet because the Sun would be so big.)

**Color:** For the Sun’s color, copy the color from the original sunAndPlanet() method. For your Planet, make it whatever color you like.

**Note:** The syntax for a color is new Color(r, g, b) where the values for r, g, b are integers between 0 and 255.

**Parameter 2: MASS**

Here we will determine the mass of your planet, relative to that of the sun. We will make the mass of your planet = 1, and mass of the Sun proportional to that.

**Raw Mass Data:**

What is the mass of your planet?

What is the mass of the sun?

**Scaled Mass Data:**

For your planet, make the mass = 1.

The actual mass of your planet will be known as the MASS SCALING FACTOR. Record it here; we will use it multiple times.

Mass Scaling Factor (MSF) =

Divide the raw mass of the Sun by the MSF. Use this number for the scaled mass of the Sun in your scenario.

Scaled mass of the Sun =

**Parameter 5 & 6: X and Y coordinates**

**X and Y Coordinates of the Sun:**

We will start by placing the Sun in the center of the scenario. You may use getWidth() and getHieght(), or the fact that the scenario is 960 x 620 pixels.

Solar X coordinate = Solar Y coordinate =

**X and Y Coordinates of the Planet**

**Raw Distance Data:**

What is the smallest distance of your planet from the Sun in km? (Planetary orbits are ellipses, not circles, so the distance is not constant.) The cosmological term for this distance is *perihelion.*

Scale this number so that the planet fits onto the scenario: When the Sun is in the center of the scenario, we want your planet to appear 300 pixels directly to the right of the Sun.

Planet X coordinate = Planet Y coordinate =

**Scaled Distance Data:**

What number would you have to divide your planet’s distance by to get the “300 pixels” you wanted? This number is the DISTANCE SCALING FACTOR (DSF). Record it here; we will use it again.

DISTANCE SCALING FACTOR (DSF) = kilometers/pixel

**Parameter 3a & 3b: Vector speed & direction**

**Vector Data:**

The Vector for the Sun should be (0, 0). (Direction = 0, Magnitude = 0). We don’t want it to move.

**Vector Direction:**

The direction of the Vector for your Planet should be straight down, which is what angle?

**Vector Length:**

The length of your vector will represent the velocity of your planet in km/hr. (Each “act” will represent 1 hour). You calculate this by the following steps:

1. Find the circumference of your planet's orbit km.
2. Calculate the number of hours your planet takes to orbit the Sun
3. Divide to calculate the velocity of your planet, in km/hr.

1. What is the circumference of the orbit of your Planet in km?

2. Calculate the number of hours it takes your planet to complete one orbit

( = number of days \* 24).

Orbit = hours

3. Divide #1 by #2 to get the velocity of your planet. (Keep at least 5 digits past the decimal)

Velocity = km/hr

3a. Divide by the DSF to get the length of your Planet’s Vector:

Scaled Velocity = pixels/act

The Scaled Velocity and Downward Direction are the two parameters of your planet’s Vector.

**The GRAVITY constant:**

Raw Number: The GRAVITY constant in our universe

= 8.6493 \* 10-13 , with units 

Convert this to your scenario by:

1. Multiplying the number above by MSF
2. Then dividing by (DSF)3

Your new GRAVITY constant is:

This number goes at the top of the Body class. There you will see a line that says:

private static final double GRAVITY = 5.8;

Replace the 5.8 with the value you arrived at.

To enter this number into your program, you will probably need the

Math.pow(base, exponent) method, which is how Java handles exponents.

For example, the number

8.644 \* 10-13 is written in Java as: 8.644 \* Math.pow(10, -13)

Put your calculate value for GRAVITY here, written in Java format:

NOTE: This value is approximate, and might not work right. Now you need to spend some time adjusting this value up or down to get it right.

Put your final value for GRAVITY here, written in Java format:

**And Now: Add a 2nd planet**

For this, you need to get the mass and distance data for the planet next to yours (either the next one inwards or outwards, whichever one will fit).

**Calculate the Mass, X and Y coordinates, and Vector for your new Planet:**

Notes:

* The new Planet should be to the LEFT of the Sun, that is, on the opposite side of the Sun from your first Planet
* The Vector’s direction for your new Planet should be straight up, since it is on the left

To find the proper values for these parameters, use the MSF, DSF, and formulas from the first part. The GRAVITY constant should not change (it is a constant, after all).

Below, show the calculations and final values for the second planet you are adding.

**Mass of the 2nd planet:**

 To get this, divide the raw mass of the second planet by the MSF.

**Vector of my 2nd planet:**

**Direction is straight down; Magnitude is calculated the same way as the first one. That is,**

1. What is the circumference of the orbit of your Planet in km?

2. Calculate the number of hours it takes your planet to complete one orbit

( = number of days \* 24).

Orbit = hours

3. Divide #1 by #2 to get the velocity of your planet. (Keep at least 5 digits past the decimal)

Velocity = km/hr

3a. Divide by the DSF to get the length of your Planet’s Vector:

Scaled Velocity = pixels/act

**X and Y coordinates:**

The Sun Body should be placed in the center of the scenario, and

Your Planet should have the same Y coordinate.

The X coordinate should be to the left of the sun, scaled by the DSF. That is

X coordinate = 480 - [Planet-to-sun distance / (DSF)]