```java
/**
 * Planet is an auxiliary class to Ringworld that allows for
 * Planets to be stored in objects.
 *
 * Group Members:
 * Luke Pastore
 * Ansh Motiani
 * Gar Rudnyai
 *
 * @author Gar Rudnyai
 * @version April 17, 2020
 */
public class Planet
{
    // Instance Variables
    private double radiusFt;
    private double volumeFt3;
    private double surfaceAreaFt2;
```

make these static

```java
    // Our Universe's Planets' Volume in Cubic Feet
    private final double JUPITER_VOLUME_FT3 = 1.43128 * Math.pow(10,15);
    private final double SATURN_VOLUME_FT3 = 8.2713 * Math.pow(10,14);
    private final double URANUS_VOLUME_FT3 = 6.833 * Math.pow(10,13);
    private final double NEPTUNE_VOLUME_FT3 = 6.254 * Math.pow(10 ,13);
    private final double EARTH_VOLUME_FT3 = 1.08321 * Math.pow(10,12);
    private final double VENUS_VOLUME_FT3 = 9.2843 * Math.pow(10,11);
    private final double MARS_VOLUME_FT3 = 1.6318 * Math.pow(10,11);
    private final double MERCURY_VOLUME_FT3 = 6.083 * Math.pow(10,10);
    private final double EARTH_SURFACE_AREA_MI2 = 196.94 * Math.pow(10, 6);

    /**
     * Planet constructor that sets the radius, volume and surface area to
     * zero
     */
    public Planet()
    {
        this.radiusFt = 0;
        this.volumeFt3 = 0;
        this.surfaceAreaFt2 = 0;
    }

    /**
     * Planet constructor that sets the radius to a given value and
     * sets the volume (ft cubed) and surface area (square feet)
     * @param radius the radius of the planet, in feet
     */
    public Planet(double radiusFt)
    {
        this.radiusFt = radiusFt;
        this.volumeFt3 = 2 * radiusFt * Math.PI;
        this.surfaceAreaFt2 = 4 * Math.pow(radiusFt, 2) * Math.PI;
    }

    /**
     * Adds a given amount of volume, assuming that the planet is
     * always perfectly spherical
     * @param double quantity in cubic feet
     */
    public void addVolume(double quantityFt3)
    {
        volumeFt3 += quantityFt3;
        radiusFt = Math.cbrt((3 / 4) * (volumeFt3 / Math.PI));
    }

    /**
     * returns volumes in miles cubed
     * @return volume in miles cubed
     */
    public double getVolume()
    {
        return Converter.ft3ToMi3(volumeFt3);
    }


    /**
     * returns surface area in square miles
     * @return surfaceArea in square miles
     */
    public double getSurfaceArea()
    {
        return Converter.ft2ToMi2(surfaceAreaFt2);
    }

    /**
    * returns array of all planet's volumes
    * @return returns array of all planet's volumes in cubic feet
    */
    public double [] getPlanets()
    {
        double [] planets = {
            JUPITER_VOLUME_FT3, SATURN_VOLUME_FT3,
            URANUS_VOLUME_FT3, NEPTUNE_VOLUME_FT3,
            EARTH_VOLUME_FT3, VENUS_VOLUME_FT3,
            MARS_VOLUME_FT3, MERCURY_VOLUME_FT3};
        return planets;
    }
}
```