Worksheet: Simple `Vector` Class

Study the final version of the `Vector` class from the previous worksheet. Then follow the steps to update the `Vector` class so that it can handle both two- and three-dimensional vectors.

```
 1 public class Vector {
 2    public double x;
 3    public double y;
 4    public Vector(double x, double y) {
 5       this.x = x;
 6       this.y = y;
 7    }
 8    public void add(double x, double y) {
 9       this.x += x;
10       this.y += y;
11    }
12    public void add(Vector v) {
13       this.x += v.x;
14       this.y += v.y;
15    }
16
17    public boolean equals(Vector v) {
18       return this.x == v.x && this.y == v.y;
19    }
20 }
```

Update the class to include:

- A third field, `z`, of type `double`.
- A second constructor that takes three parameters of type `double` in order to set all three fields of the newly created `Vector` object. The two-parameter constructor should remain in the class (i.e.: overload the constructor).
- Remove the `add` method that takes the two `double` parameters (just so you don't need to write so much).
- Update the `add` method that takes a parameter of type `Vector` so that it handles three-dimensional vectors.
- Update the `equals` method to handle three-dimensional vectors.
- Write a method `magnitude` that takes no parameters and returns the magnitude of the vector.
- Write a method `toString` that takes no parameters and returns a string representation of the vector, for example: `(0.0, 0.0, 0.0)`.