```
1
     package Version1;
 2
 3
     import java.util.ArrayList;
 4
 5
 6
     * This is the Vector class, being the first version, the methods and constructors
 7
     * are not implemented. The purpose of this version is to test it and to show the tests
8
    * created will 100% fail. Allowing us to slowly create the vector class and ensure the
9
    * class is sound.
    * /
10
11
12
    public class Vector {
13
14
         private int N = 0;
15
16
         private ArrayList<Double> data;
17
18
         //empty
19
         public Vector() {
20
             throw new UnsupportedOperationException();
21
22
23
         //a vector is created of size - size, with elements initalized to D
2.4
         public Vector(int size, double D) {
25
             throw new UnsupportedOperationException();
26
27
28
         //a vector is created to be initialized to the array D
29
         public Vector(double [] D) {
30
             throw new UnsupportedOperationException();
31
         1
32
33
         //the vector is initalized to Int I
34
         public Vector(int [] I) {
35
             throw new UnsupportedOperationException();
36
         }
37
38
         public void append(double[] doubleArray) {
39
             throw new UnsupportedOperationException();
40
         1
41
42
         public void append(int[] intArray) {
43
             throw new UnsupportedOperationException();
44
45
46
         public Vector append(Vector V) {
47
             throw new UnsupportedOperationException();
48
49
50
         public Vector append(double aDouble) {
51
             throw new UnsupportedOperationException();
52
53
54
         //this will be equivalent to the vector V
55
         public Boolean equal (Vector V) {
56
             throw new UnsupportedOperationException();
57
         }
58
59
         //returns the # of elements
60
         public int getLength() {
61
             throw new UnsupportedOperationException();
62
         }
63
64
         //returns the value this[i]
65
         public double getValue(int i) {
66
             throw new UnsupportedOperationException();
67
68
69
              //add this to V, returning a Vector the same size as this
```

```
public Vector add(Vector V) {
 71
              throw new UnsupportedOperationException();
 72
 73
 74
          //add aDouble to every element of this
 75
          public Vector add(double aDouble) {
 76
              throw new UnsupportedOperationException();
 77
 78
 79
           //sub this - V
 80
          public Vector sub(Vector V) {
 81
              throw new UnsupportedOperationException();
 82
          }
 83
 84
           //will return a sub vector between the
 85
           //indices l and r inclusive
 86
          public Vector subV(int 1, int r) {
 87
              throw new UnsupportedOperationException();
 88
          3
 89
 90
           //Multiple every element of this by corresponding element in V
 91
          public Vector Mult(Vector V) {
 92
              throw new UnsupportedOperationException();
 93
 94
 95
           //Multiply every element of this by aDouble
 96
          public Vector Mult(double aDouble) {
 97
              throw new UnsupportedOperationException();
 98
          1
 99
100
           //returns this as a normalized vector
101
          public Vector Normalize() {
102
              throw new UnsupportedOperationException();
103
          }
104
105
          //Returns the euclidean distance between this and V
106
          public double EuclidianDistance(Vector V) {
107
              throw new UnsupportedOperationException();
108
109
110
      } // end Vector
111
```