```java
/**
 * This program provides a series of tests and sample calls to demo the usage
 * and features of the ‘Box‘ class.
 *
 * @author  Ravi S. Ramphal
 * @class   CCSF CS111B
 * @date    2017.06.22
 * @version 1.0
 */

public class DemoBox
{
    /**
     * This is a helper method used to print a line above and below a given
     * message (from the middle).
     *
     * param message The message that you would like to pad
     */
    static private void mid(String message)
    {
        System.out.println(message);
        System.out.println();
    }

    /**
     * This is a helper method used to print a horizontal rule above a given
     * message.
     *
     * param message The message that you would like to preprend to
     */
    static private void pre(String message)
    {
        System.out.println();
        System.out.println("================================================");
        System.out.println();
        mid(message);
    }

    /**
     * This method takes an instance of ‘Box‘ and calls ‘@volume‘ on it
     * providing additional information on dimensions so the user can ensure
     * that the correct volume is printed for the given dimensions.
     *
     * param box An instance of ‘Box‘ that you would like to print volume for
     */
    static private void testVolume(Box box)
    {
        mid(
            "    * For a box of height " + box.height + ", width " + box.width +
            ", and depth " + box.depth + ", the volume returned is: " +
            box.volume()
        );
    }

    /**
     * This method takes an instance of ‘Box‘ and calls ‘@toString()‘ on it
     * in the context of concatenation.
     *
     * param box An instance of ‘Box‘ that you would like to cast as a string
     */
    static private void testToString(Box box)
    {
        mid("    * Created: " + box);
    }

    /**
     * This method takes an instance of ‘Box‘ and calls ‘@show()‘ on it.
     *
     * param box An instance of ‘Box‘ that you would like to call ‘show‘ on
     */
    static private void testShow(Box box)
    {
        mid("    * Info:");
        box.show();
    }

    /**
     * This method takes two instances of ‘Box‘ and calls ‘@equals()‘ to compare
     * them and prints the result.
     *
     * param box1 The first instance of ‘Box‘ that you would like to compare
     * param box2 The second instance of ‘Box‘ that you would like to compare
     */
    static private void testEquals(Box box1, Box box2)
    {
        String qualifier = (box2.equals(box1)) ? "are" : "are not";

        mid("    * " + box1 + " and " + box2 + " " + qualifier + " equal");
    }

    /**
     * This is a testing method used to create a box with three dimensions and
     * test ‘volume‘, ‘toString‘, and ‘show‘.
     */
    static private void testThreeDimensions()
    {
        pre("TESTING CREATION OF BOX WITH THREE DIMENSIONS");

        Box box = new Box(3, 6, 9);

        testVolume(box);
        testToString(box);
        testShow(box);
    }

    /**
     * This is a testing method used to create a cube and
     * test ‘@volume()‘, ‘@toString()‘, and ‘@show()‘ via helper methods.
     */
    static private void testCube()
    {
        pre("TESTING CREATION OF CUBE");

        Box box = new Box(5);

        testVolume(box);
        testToString(box);
        testShow(box);
    }

    /**
     * This is a testing method used to create a box by cloning another box and
     * test ‘@volume()‘, ‘@toString()‘, and ‘@show()‘ via helper methods. It
     * also compares it to the original via ‘@equals()‘ and compares it to
     * another cube of different dimensions.
     */
    static private void testClone()
    {
        pre("TESTING CREATION OF A BOX BY CLONING ANOTHER BOX");

        Box box1 = new Box(3);
        Box box2 = new Box(box1);
        Box box3 = new Box(4);

        testEquals(box1, box2);
        testEquals(box2, box3);
        testVolume(box2);
        testToString(box2);
        testShow(box2);
    }

    /**
     * This is a testing method used to create a box of zero dimensions using
     * the default constructor and then runs tests to
     * test ‘@volume()‘, ‘@toString()‘, and ‘@show()‘ via helper methods.
     */
    static private void testDefault()
    {
        pre("TESTING CREATION OF DEFAULT ZERO-DIMENSION BOX");

        Box box = new Box();

        testVolume(box);
        testToString(box);
        testShow(box);
    }

    /**
     * This is the main function of this demo class to call the other testing
     * methods.
     */
    public static void main(String ... args)
    {
        testThreeDimensions();
        testCube();
        testClone();
        testDefault();
    }
}
```