```
import org.junit.Before;
import org.junit.Test;

import junit.framework.TestCase;

public class ArrayFormatTest extends TestCase {
      ArrayFormat stringValue = null;
      String empty = null;
      ArrayFormat value, anotherValue = null;
      ObjectFormat obj, anotherObj = null;

      @Before
      public void setUp() {
            stringValue = ArrayFormat.fromString(
                        "[10.0,\"Design Patterns
\",[190,1.2,89,1100.0,\"string\",90],1980,{\"string\":\"String
value\",\"one\":1,\"array\":[1,1.2,1800.0,\"string\"]}]");
            value = new ArrayFormat();
            anotherValue = new ArrayFormat();
            anotherValue.put(12).put("read");
            obj = new ObjectFormat();
            anotherObj = new ObjectFormat();
            obj.put("key", "value");
            value.put(12).put("cs696").put(1678.90).put(obj);
            anotherObj.put("integer", 678).put("string",
"summer").put("double", 189.789).put("array", anotherValue);
      }

      @Test
      public void testArrayFormat() throws
ArrayIndexOutOfBoundsException, DataTypeMisMatchException {
            System.out.println("Running testArrayFormat()");

      stringValue.put(167.90).put(78).put("Assignment3").put(value
);
            assertEquals(value, stringValue.getArray(8));
            stringValue.put(obj);
            assertEquals(obj, stringValue.getObject(9));
            assertEquals("Design Patterns",
stringValue.getString(1));
            assertEquals(10.0, stringValue.getDouble(0));
            assertEquals(1980, stringValue.getInt(3));
            try {
                  stringValue.getArray(1);
                  fail("Expected an DataTypeMisMatchexception to be
thrown");
            } catch (DataTypeMisMatchException
anDataTypeMisMatchException) {

      assertEquals(anDataTypeMisMatchException.getMessage(),
"value is not of array type");
            }
```

```
          try {
                stringValue.put(empty);
                fail("Expected an IllegalArgumentException to be
thrown");
          } catch (IllegalArgumentException
anIllegalArgumentException) {

     assertEquals(anIllegalArgumentException.getMessage(), "Value
cannot be null");
          }
          assertEquals(10, stringValue.length());
          assertEquals("Design Patterns",
stringValue.remove(1));
          try {
                stringValue.get(23);
                fail("Expected an ArrayIndexOutOfBoundsException
to be thrown");
          } catch (ArrayIndexOutOfBoundsException
anArrayIndexOutOfBoundsException) {

     assertEquals(anArrayIndexOutOfBoundsException.getMessage(),
"Invalid index");
          }
          assertEquals(4, value.length());
          value.put("history");
          assertEquals(5, value.length());
          assertEquals(12, value.remove(0));
          assertEquals(obj, value.remove(2));
          Object[] expectedOutput = new Object[] { "cs696",
1678.9, "history" };
          assertTrue("Arrays not the same length",
value.length() == expectedOutput.length);
          for (int i = 0; i < value.length(); i++)
                assertEquals(expectedOutput[i], value.get(i));
          assertEquals("[12,\"read\"]",
anotherValue.toString());
     }
}
```