2. JUnit program that demonstrates assertions:

import org.junit.\*;

public class AssertionTest {

@Test

public void testAssertEquals() {

String expected = "Hello, World!";

String actual = "Hello, World!";

Assert.assertEquals(expected, actual);

}

@Test

public void testAssertTrue() {

boolean condition = true;

Assert.assertTrue(condition);

}

@Test

public void testAssertFalse() {

boolean condition = false;

Assert.assertFalse(condition);

}

@Test

public void testAssertNull() {

Object obj = null;

Assert.assertNull(obj);

}

@Test

public void testAssertNotNull() {

Object obj = new Object();

Assert.assertNotNull(obj);

}

@Test

public void testAssertSame() {

Object obj1 = new Object();

Object obj2 = obj1;

Assert.assertSame(obj1, obj2);

}

@Test

public void testAssertNotSame() {

Object obj1 = new Object();

Object obj2 = new Object();

Assert.assertNotSame(obj1, obj2);

}

}

In this example, we have defined a JUnit test class called AssertionTest. This class contains several test methods that demonstrate different types of assertions:

* testAssertEquals: This method uses the assertEquals method to compare two strings and check if they are equal.
* testAssertTrue: This method uses the assertTrue method to check if a boolean condition is true.
* testAssertFalse: This method uses the assertFalse method to check if a boolean condition is false.
* testAssertNull: This method uses the assertNull method to check if an object is null.
* testAssertNotNull: This method uses the assertNotNull method to check if an object is not null.
* testAssertSame: This method uses the assertSame method to check if two objects are the same instance.
* testAssertNotSame: This method uses the assertNotSame method to check if two objects are not the same instance.