Exercise 1: Setting Up JUnit:

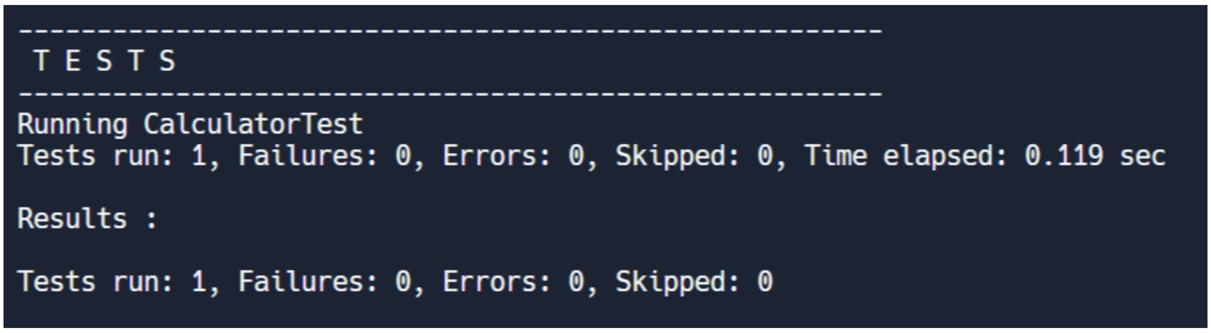
Calculator.java:

```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
}
```

CalculatorTest.java:

```
import org.junit.Test;
import static org.junit.Assert.assertEquals;
public class CalculatorTest {
    @Test
    public void testAdd() {
        Calculator calculator = new Calculator();
        int result = calculator.add(5,4);
        assertEquals(9, result);
    }
}
```

Output:



```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running CalculatorTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.119 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

Exercise 2: Writing Basic JUnit Tests:

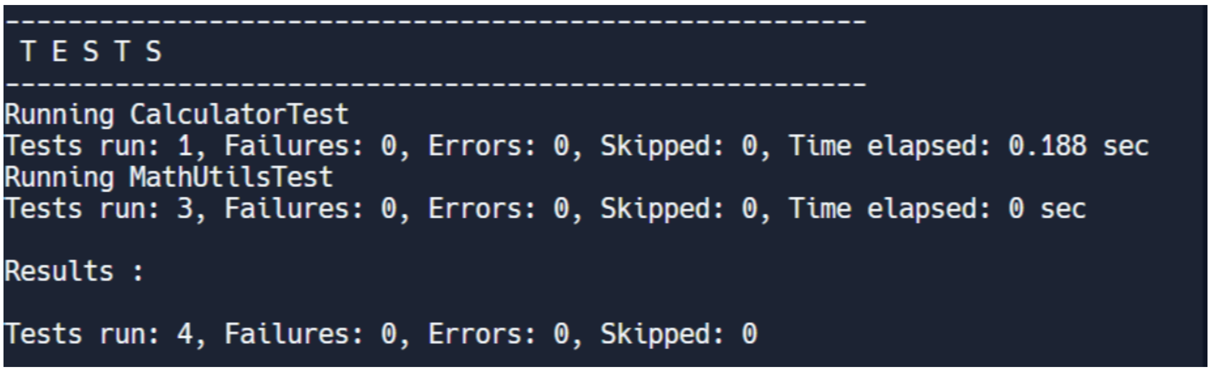
MathUtils.java:

```
public class MathUtils {
    public int add(int a, int b) {
        return a + b;
    }
    public int multiply(int a, int b) {
        return a * b;
    }
    public boolean isEven(int number) {
        return number % 2 == 0;
    }
}
```

MathUtilsTest.java:

```
import org.junit.Test;
import static org.junit.Assert.*;
public class MathUtilsTest {
    @Test
    public void testAdd() {
        MathUtils mathUtils = new MathUtils();
        int result = mathUtils.add(5, 4);
        assertEquals(9, result);
    }
    @Test
    public void testMultiply() {
        MathUtils mathUtils = new MathUtils();
        int result = mathUtils.multiply(2, 8);
        assertEquals(16, result);
    }
    @Test
    public void testIsEven() {
        MathUtils mathUtils = new MathUtils();
        assertTrue(mathUtils.isEven(20));
        assertFalse(mathUtils.isEven(31));
    }
}
```

Output:



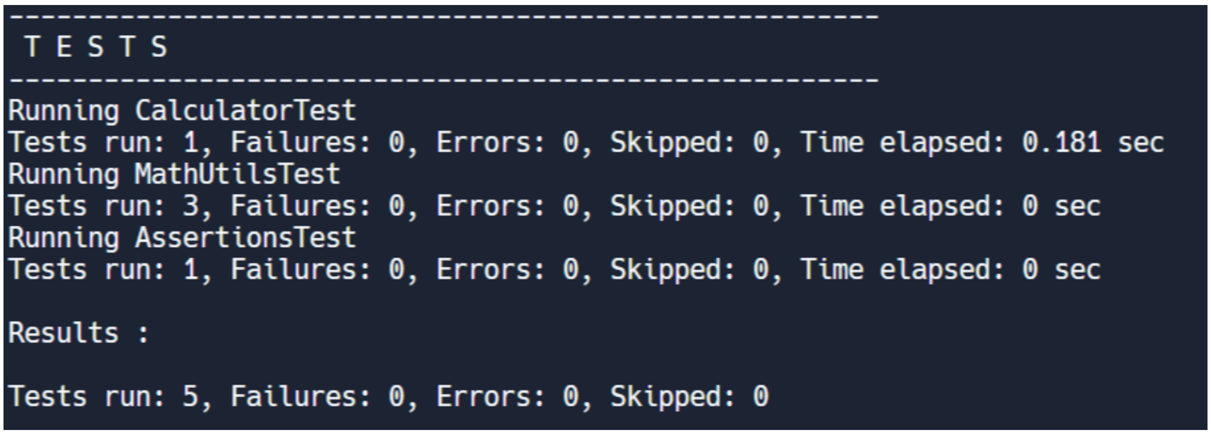
```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running CalculatorTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.188 sec
Running MathUtilsTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec

Results :

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
```

Exercise 3: Assertions in JUnit:

AssertionsTest.java:

```
import org.junit.Test;
import static org.junit.Assert.*;
public class AssertionsTest {
    @Test
    public void testAssertions() {
        assertEquals(3, 1 + 2);
        assertTrue(7 > 1);
        assertFalse(7 < 1);
        Object obj1 = null;
        assertNull(obj1);
        Object obj2 = new Object();
        assertNotNull(obj2);
    }
}
```

Output:



```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running CalculatorTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.181 sec
Running MathUtilsTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec
Running AssertionsTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
```

Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit:

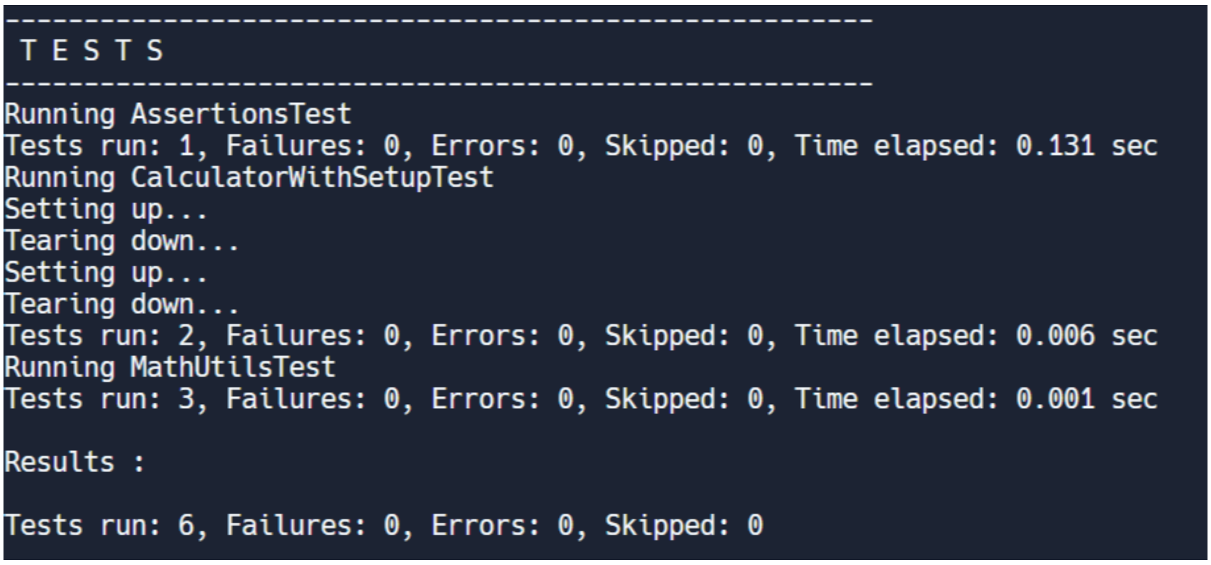
Calculator.java:

```
public class Calculator {
   public int add(int a, int b) {
      return a + b;
   }
   public int subtract(int a, int b) {
      return a - b;
   }
}
```

CalculatorTest.java:

```
import org.junit.Before;
import org.junit.After;
import org.junit.Test;
import static org.junit.Assert.*;
public class Calculatortest {
   private Calculator calculator;
   @Before
   public void setUp() {
      System.out.println("Setting up...");
      calculator = new Calculator();
   }
   @After
   public void tearDown() {
      System.out.println("Tearing down...");
      calculator = null;
   }
   @Test
   public void testAdd() {
      int a = 5;
      int b = 3;
      int result = calculator.add(a, b);
      assertEquals(8, result);
   }
   @Test
   public void testSubtract() {
      int a = 10;
      int b = 4;
      int result = calculator.subtract(a, b);
      assertEquals(6, result);
   }
}
```

Output:



```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running AssertionsTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.131 sec
Running CalculatorWithSetupTest
Setting up...
Tearing down...
Setting up...
Tearing down...
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 sec
Running MathUtilsTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.001 sec

Results :

Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
```