

JUnit 5 and Mockito 3 - PART-1

```
<dependencies>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <version>5.6.2</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-core</artifactId>
    <version>3.3.3</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-junit-jupiter</artifactId>
    <version>3.3.3</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

Normal classes:

```
package in.nit;

public interface Maths {

    public int add(int a,int b) ;

}

package in.nit.service;

import in.nit.Maths;

public class ProductCatalog {

    private Maths m;

    public ProductCatalog(Maths m) {
```

```
        this.m=m;
    }

    public String getMsg(int len,int wid) {
        return "Hello Data is:"+m.add(len, wid);
    }
}
```

UnitTest files:-

```
package in.nit.test;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.when;

import java.util.Iterator;

import org.junit.jupiter.api.Test;

import in.nit.Maths;

public class TestMock {

    @Test
    public void testMockA() {
        int a=10,b=20;
        Maths m=mock(Maths.class);
        when(m.add(a, b)).thenReturn(a+b);
        assertEquals(30, m.add(a, b));
    }

    @Test
    public void testMockB() {
        Iterator<String> itr=mock(Iterator.class);
        when(itr.next()).thenReturn("One").thenReturn("Two");

        String actual=itr.next()+" "+itr.next();
        assertEquals("One Two", actual);
    }

    @Test
    public void testMockC() {
```

```
        Comparable<String> comp=mock(Comparable.class);
        when(comp.compareTo("RAGHU")).thenReturn(10);
        when(comp.compareTo("SIR")).thenReturn(20);

        assertEquals(10, comp.compareTo("RAGHU"));
    }
}

package in.nit.test;

import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.mockito.Mockito.when;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Disabled;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.MethodOrderer.OrderAnnotation;
import org.junit.jupiter.api.Order;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.TestMethodOrder;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;

import in.nit.Maths;
import in.nit.service.ProductCatalog;

@DisplayName("Test Maths Operations")
@TestMethodOrder(OrderAnnotation.class)
public class TestMath {

    @Mock
    private Maths m;
    private ProductCatalog pc;

    private int a,b;
    String expected,actual;

    @BeforeEach
    public void doSetup() {
        MockitoAnnotations.initMocks(this);
        expected="Hello Data is: "+(a+b);
    }
}
```

```
@DisplayName("Test Add One")
@Test
@Order(4)
public void testAdd() {
    when(m.add(a, b)).thenReturn(a+b);

    int actual=m.add(a, b);
    assertEquals(a+b, actual);
}

@DisplayName("Test Add Two")
@Test
@Order(2)
@Disabled
public void testAddService() {
    when(m.add(a, b)).thenReturn(a+b);

    pc=new ProductCatalog(m);
    actual=pc.getMsg(10, 20);
    assertEquals(expected, actual);
}

@AfterEach
public void close() {
    m=null;
}
}
```