**SOURCE CODE**

**Authentication.class**

package test10;

public class Authentication {

public boolean Validate(String username, String Password)

{

if(username.equals("Pradeepa") && Password.equals("1234"))

return true;

else return false;

}

}

**DependencyInjection.java**

package test10;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Tag;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.extension.ExtendWith;

@ExtendWith(ValidateParameterResolver.class)

class DI3 {

@Test

@Tag("my-tag")

void test( Authentication at) {

assertTrue(at.Validate("Pradeepa", "1234"));

}

}

**ValidateParameterResolver.class**

package test10;

import static org.junit.jupiter.api.Assertions.\*;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.extension.ExtensionContext;

import org.junit.jupiter.api.extension.ParameterContext;

import org.junit.jupiter.api.extension.ParameterResolutionException;

import org.junit.jupiter.api.extension.ParameterResolver;

class ValidateParameterResolver implements ParameterResolver {

@Override

public boolean supportsParameter(ParameterContext parameterContext, ExtensionContext extensionContext)

throws ParameterResolutionException {

return parameterContext.getParameter().getType()==Authentication.class;

}

@Override

public Object resolveParameter(ParameterContext parameterContext, ExtensionContext extensionContext)

throws ParameterResolutionException {

return new Authentication();

}

}