**Day-13 Basics of Java**

Problem Statement 1: Setting Up, Writing, and Running Java Unit Tests

**package** Junit;

**import** **static** org.junit.Assert.\*;

**import** java.sql.Connection;

**import** java.sql.SQLException;

**import** java.util.List;

**import** org.junit.After;

**import** org.junit.AfterClass;

**import** org.junit.Before;

**import** org.junit.BeforeClass;

**import** org.junit.Test;

**import** Exercise\_day12.DBUtil;

**import** Exercise\_day12.Product;

**import** Exercise\_day12. ProductManagementDAO;

**public** **class** ProductManagementTest {

**private** **static** Connection *connection*;

**private** ProductManagementDao.dao;

@BeforeClass

**public** **static** **void** init() **throws** Exception {

*connection* = DBUtil.getConnection();

}

@Before

**public** **void** beforeEachTest(){

dao = **new** ProductManagement.DAO();

}

@After

**public** **void** tearDown() **throws** Exception {

}

@Test

**public** **void** addProductTest() **throws** SQLException {

  Product p=**new** Product(8, "snacks", 170.0);

dao.addProduct (p);

             Product newproduct=dao.getProductByld(8);

assertNotNull("product should not be null",newprod);

assertEquals("product name should be match", "snacks", newprod.getName());

assertEquals("product price should be match", 170.0, newprod.getPrice(),0.01);

}

@Test

**public** **void** getAllProductsTest() **throws** SQLException {

List<Product> products = dao.getAllProducts();

assertNotNull("Products list should not be null", products);

assertTrue("Products list should not be empty", products.size()>0);

}

@Test

**public** **void** deleteProductTest() **throws** SQLException {

dao.deleteProduct(4);

Product delProd = dao.getProductById(4);

assertNull("Deleted product should be null", delProd);

}

@Test

**public** **void** updateProductTest() **throws** SQLException {

Product p=**new** Product(2, "clock", 200.0);

dao.updateProduct(p);

Product newprod = dao.getProductById(2);

assertEquals("product name should be match", "clock", newprod.getName());

assertEquals("product proce should be match", 200.0, newprod.getPrice(),0.01);

}

@AfterClass

**public** **static** **void** destroy() {

**if**(*connection* != **null**) {

**try** {

*connection*.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

}