**Problem Statement 1: Setting Up, Writing, and Running Java**

**Unit Tests**

**import** com.mphasis.dao.\*;

**import** com.mphasis.app.\*;

**import** com.mphasis.dbutil.\*;

**import** com.mphasis.domain.Product;

**import** org.junit.\*;

**import** java.sql.Connection;

**import** java.sql.SQLException;

**import** java.util.List;

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

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

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

**private** ProductDAO productDAO;

    @BeforeClass

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

*connection* = DBUtil.*getConnection*();

        System.***out***.println("Connection established.");

    }

    @Before

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

        productDAO = **new** ProductDAO();

    }

    @Test

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

        Product product = **new** Product("Test Product", 999.99);

**int** isAdded = productDAO.addProduct(product);

*assertEquals*(1, isAdded);

    }

    @Test

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

        List<Product> products = productDAO.viewProducts();

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

*assertTrue*("Product list should contain at least one product", products.size() > 0);

    }

    @Test

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

        Product product = **new** Product("Temp Product", 500.00);

        productDAO.addProduct(product);

        List<Product> products = productDAO.viewProducts();

**int** productIdToDelete = products.get(products.size() - 1).getProductID();

**int** isDeleted = productDAO.deleteProduct(productIdToDelete);

*assertEquals*(0,isDeleted);

    }

    @Test

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

        Product product = **new** Product("Update Test Product", 123.45);

        productDAO.addProduct(product);

        List<Product> products = productDAO.viewProducts();

**int** productIdToUpdate = products.get(products.size() - 1).getProductID();

        Product updatedProduct = **new** Product(productIdToUpdate, "Updated Product Name", 678.90);

**int** isUpdated = productDAO.updateProduct(updatedProduct);

*assertEquals*(0, isUpdated);

        Product fetchedProduct = productDAO.searchProduct(productIdToUpdate);

    }

    @AfterClass

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

**if** (*connection* != **null** && !*connection*.isClosed()) {

*connection*.close();

        }

        System.***out***.println("Connection closed.");

    }

}