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

**BankAccount.java**

class BankAccount {  
 private int balance;  
  
 public BankAccount(int initialBalance) {  
 this.balance = initialBalance;  
 }  
  
 public void deposit(int amount) {  
 if (amount > 0) balance += amount;  
 }  
  
 public void withdraw(int amount) {  
 if (amount <= balance) balance -= amount;  
 }  
  
 public int getBalance() {  
 return balance;  
 }  
}

**BankAccountTest.java**

import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;

public class BankAccountTest {  
  
 private BankAccount account;  
  
 @Before  
 public void setUp() {  
 account = new BankAccount(100);  
 System.*out*.println("Setup completed.");  
 }  
  
  
 @After  
 public void tearDown() {  
 account = null;  
 System.*out*.println("Teardown completed.");  
 }  
  
 @Test  
 public void testDeposit() {  
  
 account.deposit(50);  
  
  
 *assertEquals*("Deposit failed", 150, account.getBalance());  
 }  
  
 @Test  
 public void testWithdrawWithinBalance() {  
 // Act  
 account.withdraw(40);  
  
 // Assert  
 *assertEquals*("Withdraw did not subtract correctly", 60, account.getBalance());  
 }  
  
 @Test  
 public void testWithdrawMoreThanBalance() {  
  
 account.withdraw(200);  
  
  
 *assertEquals*("Over-withdrawal should not change balance", 100, account.getBalance());  
 }  
}

**Output:**

