

COMSATS University Islamabad Abbottabad campus

Lab Final

SUBJECT:	Software Testing
SUBMITTED TO:	Sir Mazhar Bukhari
SUBMITTED BY:	Najeeb Said
REGISTRATION NO:	FA20-BSE-023
DATE:	Tuesday, 11 July 2023


Q1:

Test ID	Description	Input	Expected Result	Actual Result	Verdict
TC-1	To Add Employee Full Name	<ul style="list-style-type: none">• Enter firstName in first input field• Enter middleName in middle input field• Enter lastName in last input field	Name Should Change Successfully	Successfully Updated	Pass
TC-2	To check whether empty name is accepted	Enter nothing in all the fields.	An error message saying name is required should be displayed	Required	Pass
TC-3	To check duplicate employee id is accepted	Enter 0034 in employee id field.	An error message saying duplicate employee id	Employee ID already Exists	Pass
TC-4	To check a file greater than 1 Mb	Add a file of size >1MB	File Size Exceeds the limit	Attachment Size Exceeded	Pass
TC-5	To check a file with valid size	Add a file less than 1 MB size	File should be accepted	Successfully Saved	Pass
TC-6	To check whether a file description is editable	<ul style="list-style-type: none">• Press the pencil icon on one of file• Add or edit something in description box• Save	New Description should be edited and visible in summary	Successfully Updated	Pass

Q2:

- Identify and describe the bug in the code.

The Bug is in the calculate(n) method. The marked line has an error

```
public static int calculate(int num) {  
    int result = 0;  
    for (int i = 1; i < num; i++) {  
        if (i % 2 == 0) {  
            result += i;  
        } else {  
            result += i;   
        }  
    }  
    return result;  
}
```

- Explain the expected behaviour of the code.

This code calculates the sum of numbers from 1 to a given value (n) and then performs a complex calculation based on the calculate method. The calculate method calculates the sum of even numbers and subtracts the sum of odd numbers from the result. The output of the program includes the sum of numbers from 1 to n and the result of the complex calculation.

- Provide the corrected version of the code.

```
public class FinalExamDebugging {  
    public static void main(String[] args) {  
        int n = 9;  
        int sum = 0;  
        for (int i = 1; i <= n; i++) {  
            sum += i;  
        }  
        System.out.println("The sum of numbers from 1 to " + n + " is: " +  
sum);  
        System.out.println("Executing a complex calculation...");  
        int result = calculate(n);  
        System.out.println("The result of the calculation is: " +  
result);  
    }  
    public static int calculate(int num) {  
        int result = 0;  
        for (int i = 1; i < num; i++) {  
            if (i % 2 == 0) {  
                result += i;  
            } else {  
                result -= i;  
            }  
        }  
        return result;  
    }  
}
```

- Explain the changes made to fix the bug.

The changes were made in else statement inside the for loop of calculate function. The statement `result+=i;` was changed to `result-=i;`

- Verify the correctness of the corrected code by providing the expected output.

The expected output for n=9 is sum=45 and result=4.

Before Bug Fix:

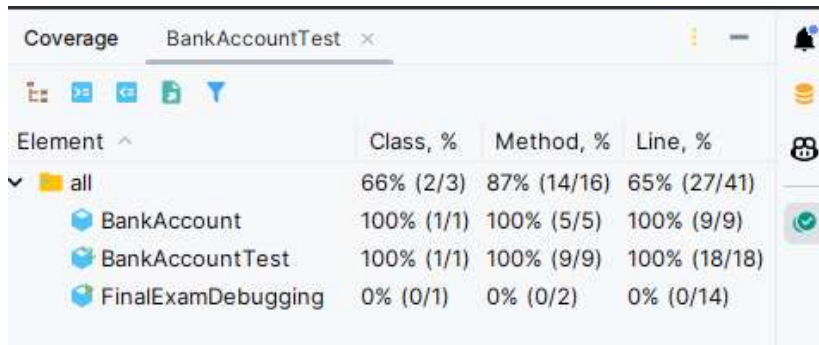
```
"C:\Program Files\Java\jdk-19\bin\java.exe" ...  
Connected to the target VM, address: '127.0.0.1:64337', transport: 'socket'  
The sum of numbers from 1 to 9 is: 45  
Executing a complex calculation...  
The result of the calculation is: 36  
Disconnected from the target VM, address: '127.0.0.1:64337', transport: 'socket'  
  
Process finished with exit code 0
```

After Bug Fix:

```
"C:\Program Files\Java\jdk-19\bin\java.exe" ...  
Connected to the target VM, address: '127.0.0.1:64315', transport: 'socket'  
The sum of numbers from 1 to 9 is: 45  
Executing a complex calculation...  
The result of the calculation is: 4  
Disconnected from the target VM, address: '127.0.0.1:64315', transport: 'socket'  
  
Process finished with exit code 0
```

Q3:

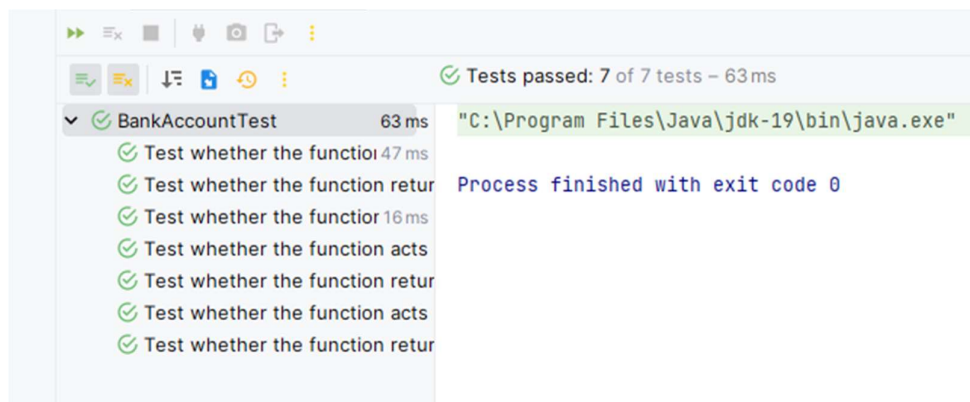
Coverage:



The screenshot shows the Coverage window in an IDE for the test class BankAccountTest. It displays a table with coverage metrics for the entire test suite and individual classes.

Element	Class, %	Method, %	Line, %
all	66% (2/3)	87% (14/16)	65% (27/41)
BankAccount	100% (1/1)	100% (5/5)	100% (9/9)
BankAccountTest	100% (1/1)	100% (9/9)	100% (18/18)
FinalExamDebugging	0% (0/1)	0% (0/2)	0% (0/14)

Test Results:



The screenshot shows the Test Results window in an IDE. It indicates that all 7 tests passed successfully within 63 milliseconds. The test runner used is Java 19.

Tests passed: 7 of 7 tests – 63 ms

BankAccountTest 63 ms

- Test whether the function returns the correct account number 47 ms
- Test whether the function returns the correct account number 16 ms
- Test whether the function acts as expected
- Test whether the function returns the correct account number
- Test whether the function acts as expected
- Test whether the function returns the correct account number

Process finished with exit code 0

Code:

```
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class BankAccountTest {
    BankAccount bankAccount;

    @BeforeEach
    void setUp() {
        bankAccount = new BankAccount("123456789");
    }

    @AfterEach
    void tearDown() {
        bankAccount = null;
    }

    @Test
    @DisplayName("Test whether the function returns the correct account number")
```

```

void TestGetAccountNumber() {
    assertEquals("123456789", bankAccount.getAccountNumber());
}

@Test
@DisplayName("Test whether the function returns the correct balance")
void TestGetBalance() {
    assertEquals(0.0, bankAccount.getBalance());
}

@Test
@DisplayName("Test whether the function returns the correct balance
after deposit")
void TestDeposit() {
    bankAccount.deposit(100);
    assertEquals(100, bankAccount.getBalance());
}

@Test
@DisplayName("Test whether the function returns the correct balance
after withdraw")
void TestWithdraw() {
    bankAccount.deposit(100);
    bankAccount.withdraw(50);
    assertEquals(50, bankAccount.getBalance());
}

@Test
@DisplayName("Test whether the function acts correctly when the
withdraw amount is negative")
void TestNegativeDeposit() {
    bankAccount.deposit(-100);
    assertEquals(0, bankAccount.getBalance());
}

@Test
@DisplayName("Test whether the function acts correctly when the
withdraw amount is negative")
void TestNegativeWithdraw() {
    bankAccount.deposit(100);
    bankAccount.withdraw(-50);
    assertEquals(100, bankAccount.getBalance());
}

@Test
@DisplayName("Test whether the function acts correctly when the
withdraw amount is above balance")
void TestWithdrawAboveBalance() {
    bankAccount.deposit(100);
    bankAccount.withdraw(200);
    assertEquals(100, bankAccount.getBalance());
}
}

```

Results:



Test Results -
BankAccountTest.ht

Q4:

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import static java.lang.Thread.sleep;

public class AmazonTest {
    private WebDriver driver;

    AmazonTest() {
        SetUp();
    }

    private void SetUp() {
        System.setProperty("webdriver.chrome.driver",
"F:\\chromedriver_win32\\chromedriver.exe");
        driver=new ChromeDriver();
    }

    public void TestLogin() {
        driver.get("https://www.amazon.com/");
        driver.findElement(By.id("nav-link-accountList")).click();
        driver.findElement(By.id("ap_email")).sendKeys("tcanjb@gmail.com");
        driver.findElement(By.id("continue")).click();

driver.findElement(By.id("ap_password")).sendKeys("Pu)yAhPQ9Rd:m2W");
        driver.findElement(By.id("signInSubmit")).click();
        TestClose();
    }

    public void TestSearch(){
        driver.get("https://www.amazon.com/");

driver.findElement(By.id("twotabsearchtextbox")).sendKeys("iphone");
        driver.findElement(By.id("nav-search-submit-button")).click();
        TestClose();
    }

    public void TestGoToCart(){
        driver.get("https://www.amazon.com/");

driver.findElement(By.id("twotabsearchtextbox")).sendKeys("iphone");
        driver.findElement(By.id("nav-search-submit-button")).click();
        driver.findElement(By.id("nav-cart")).click();
        TestClose();
    }

    public void TestLogout() {
        driver.get("https://www.amazon.com/");
        driver.findElement(By.id("nav-link-accountList")).click();
        driver.findElement(By.id("nav-item-signout")).click();
        TestClose();
    }

    public void TestClose() {
        try {
            sleep(2000);
            driver.close();
        }
    }
}
```

```
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
}  
}
```