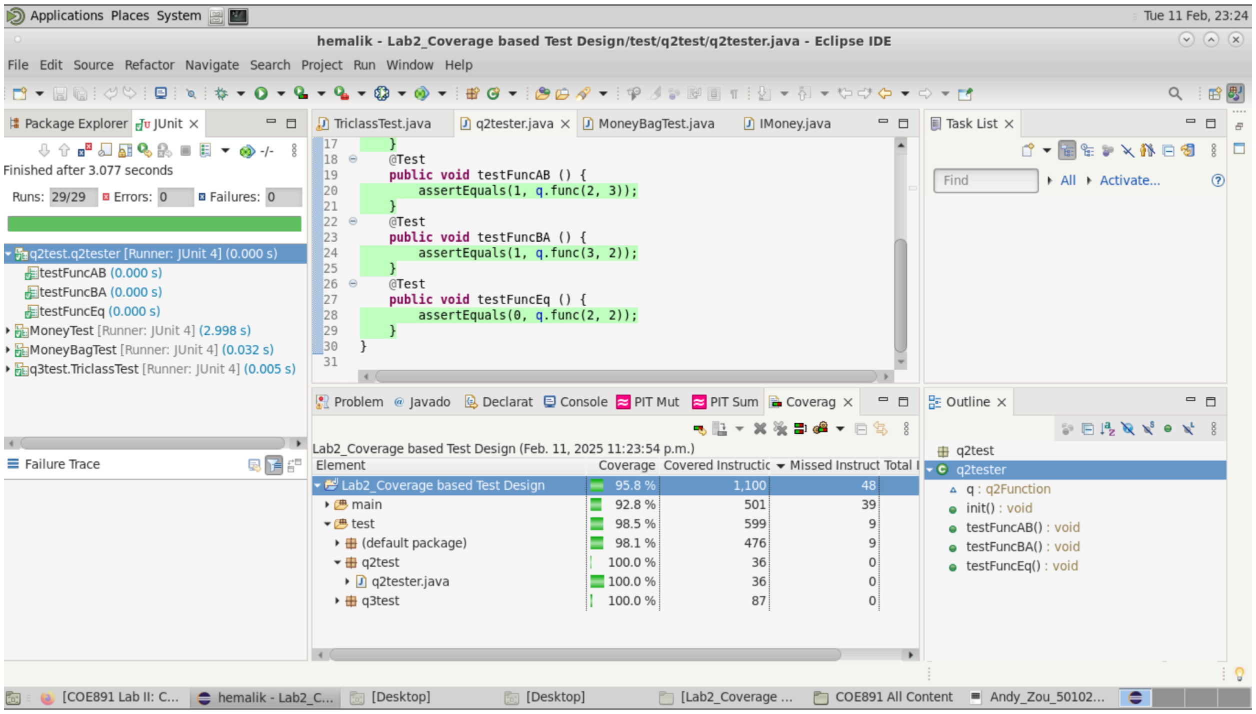
COE891 Lab II: Coverage-based Test Design

Hamza Malik
501112545

Feb 11, 2025



```
Finished after 3.077 seconds
Runs: 29/29   Errors: 0   Failures: 0

q2test.q2tester [Runner: JUnit 4] (0.000 s)
    testFuncAB (0.000 s)
    testFuncBA (0.000 s)
    testFuncEq (0.000 s)
MoneyTest [Runner: JUnit 4] (2.998 s)
MoneyBagTest [Runner: JUnit 4] (0.032 s)
q3test.TriclassTest [Runner: JUnit 4] (0.005 s)
```

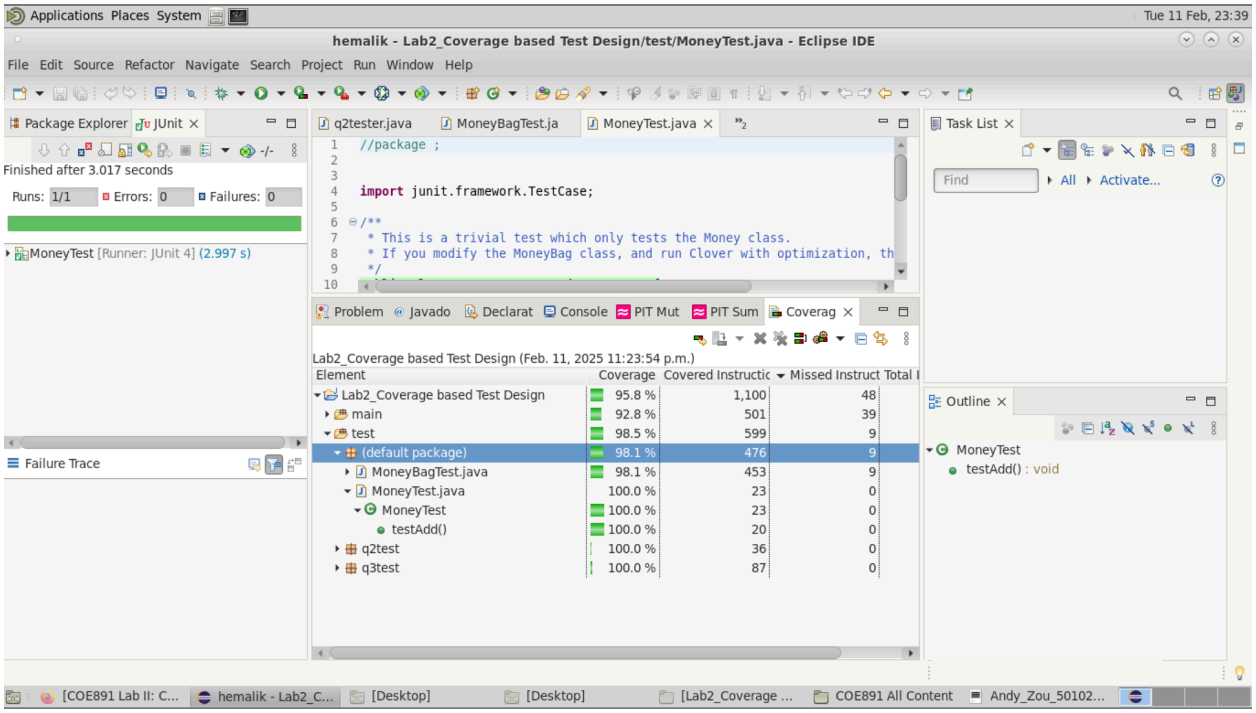
```java
    }
    @Test
    public void testFuncAB () {
        assertEquals(1, q.func(2, 3));
    }
    @Test
    public void testFuncBA () {
        assertEquals(1, q.func(3, 2));
    }
    @Test
    public void testFuncEq () {
        assertEquals(0, q.func(2, 2));
    }
}
```

Lab2_Coverage based Test Design (Feb. 11, 2025 11:23:54 p.m.)

Element	Coverage	Covered Instructic	Missed Instruct	Total I
Lab2_Coverage based Test Design	95.8 %	1,100	48	
main	92.8 %	501	39	
test	98.5 %	599	9	
(default package)	98.1 %	476	9	
q2test	100.0 %	36	0	
q2tester.java	100.0 %	36	0	
q3test	100.0 %	87	0	

```
q2test
q2tester
    q : q2Function
    init() : void
    testFuncAB() : void
    testFuncBA() : void
    testFuncEq() : void
```

After running all test suites, including q2test, TriclassTest, MoneyBagTest, and MoneyTest, the overall test coverage reached 95.8%. While q2test and TriclassTest achieved full statement coverage, MoneyBagTest had minor uncovered areas. These results demonstrate that our test design effectively validates the correctness of the implemented functionalities, with slight improvements needed to achieve complete coverage across all components



hemalik - Lab2_Coverage based Test Design/test/MoneyTest.java - Eclipse IDE

Tue 11 Feb, 23:39

Finished after 3.017 seconds

Runs: 1/1 Errors: 0 Failures: 0

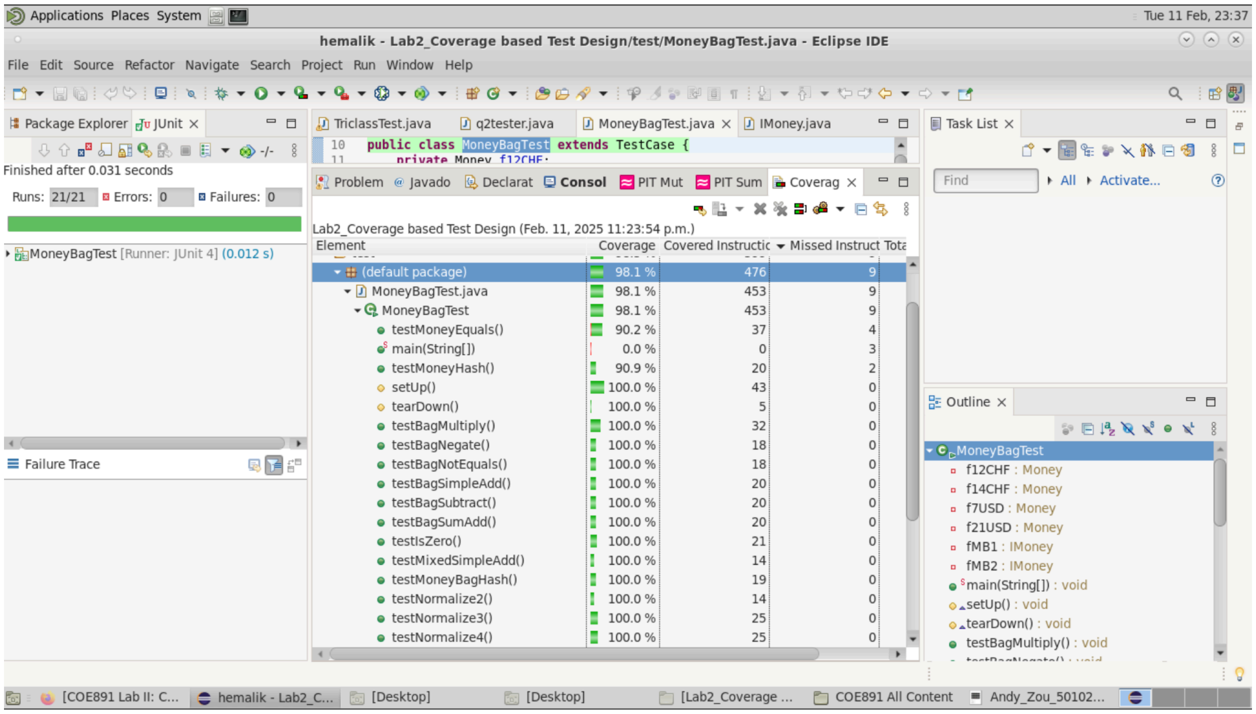
MoneyTest [Runner: JUnit 4] (2.997 s)

```
//package ;


import junit.framework.TestCase;

/**
 * This is a trivial test which only tests the Money class.
 * If you modify the MoneyBag class, and run Clover with optimization, th
 */
```

Lab2_Coverage based Test Design (Feb. 11, 2025 11:23:54 p.m.)

Element	Coverage	Covered Instructic	Missed Instruct
Lab2_Coverage based Test Design	95.8 %	1,100	48
main	92.8 %	501	39
test	98.5 %	599	9
(default package)	98.1 %	476	9
MoneyBagTest.java	98.1 %	453	9
MoneyTest.java	100.0 %	23	0
MoneyTest	100.0 %	23	0
testAdd()	100.0 %	20	0
q2test	100.0 %	36	0
q3test	100.0 %	87	0

Outline: MoneyTest
testAdd() : void



hemalik - Lab2_Coverage based Test Design/test/MoneyBagTest.java - Eclipse IDE

Tue 11 Feb, 23:37

Finished after 0.031 seconds

Runs: 21/21 Errors: 0 Failures: 0

MoneyBagTest [Runner: JUnit 4] (0.012 s)

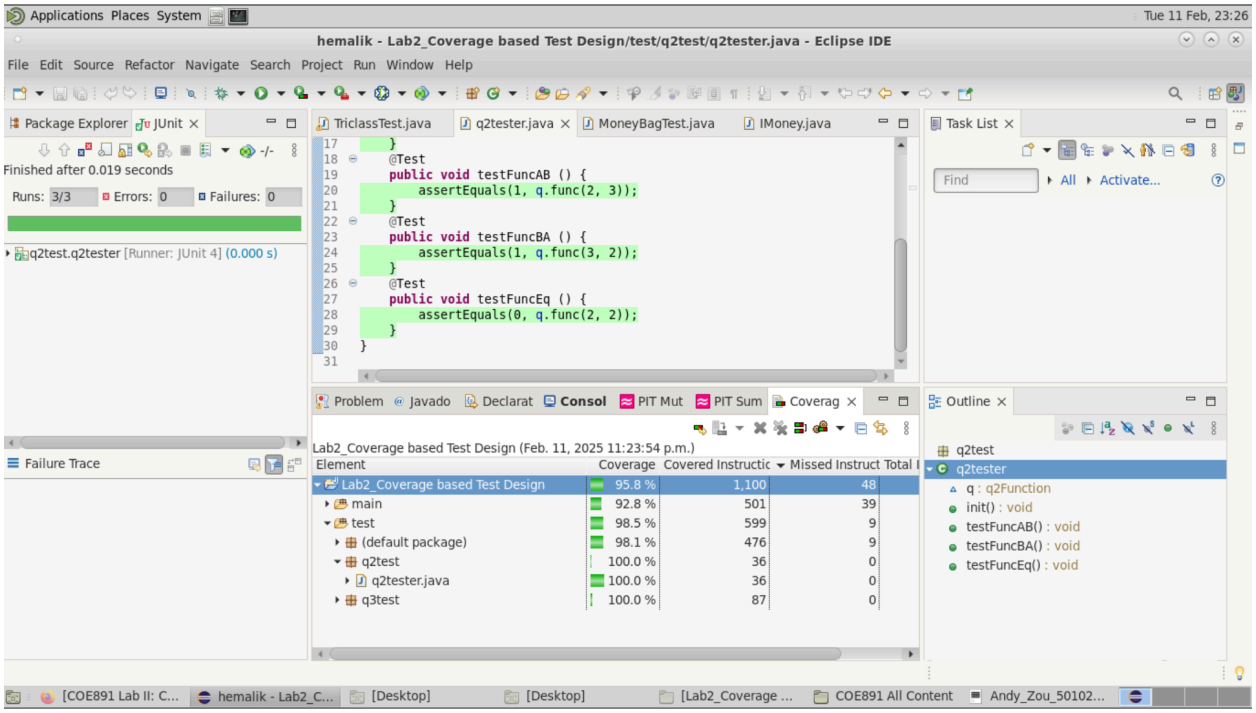
```
public class MoneyBagTest extends TestCase {
    private Money f12CHF;
```

Lab2_Coverage based Test Design (Feb. 11, 2025 11:23:54 p.m.)

Element	Coverage	Covered Instructic	Missed Instruct
(default package)	98.1 %	476	9
MoneyBagTest.java	98.1 %	453	9
MoneyBagTest	98.1 %	453	9
testMoneyEquals()	90.2 %	37	4
main(String[])	0.0 %	0	3
testMoneyHash()	90.9 %	20	2
setUp()	100.0 %	43	0
tearDown()	100.0 %	5	0
testBagMultiply()	100.0 %	32	0
testBagNegate()	100.0 %	18	0
testBagNotEquals()	100.0 %	18	0
testBagSimpleAdd()	100.0 %	20	0
testBagSubtract()	100.0 %	20	0
testBagSumAdd()	100.0 %	20	0
testIsZero()	100.0 %	21	0
testMixedSimpleAdd()	100.0 %	14	0
testMoneyBagHash()	100.0 %	19	0
testNormalize2()	100.0 %	14	0
testNormalize3()	100.0 %	25	0
testNormalize4()	100.0 %	25	0

Outline: MoneyBagTest
f12CHF : Money
f14CHF : Money
f7USD : Money
f21USD : Money
fMB1 : IMoney
fMB2 : IMoney
main(String[]) : void
setUp() : void
tearDown() : void
testBagMultiply() : void

Q1: The MoneyBagTest suite was executed to verify various monetary operations such as addition, subtraction, and equality comparisons. Initially, some methods had lower coverage, so additional test cases were introduced to address missing execution paths. The final coverage report shows 98.1% statement coverage, ensuring that almost all statements were exercised during testing. While a small gap remains, the overall results confirm that the majority of the code is well-tested.



hemalik - Lab2_Coverage based Test Design/test/q2test/q2tester.java - Eclipse IDE

Finished after 0.019 seconds

Runs: 3/3 Errors: 0 Failures: 0

q2test.q2tester [Runner: JUnit 4] (0.000 s)

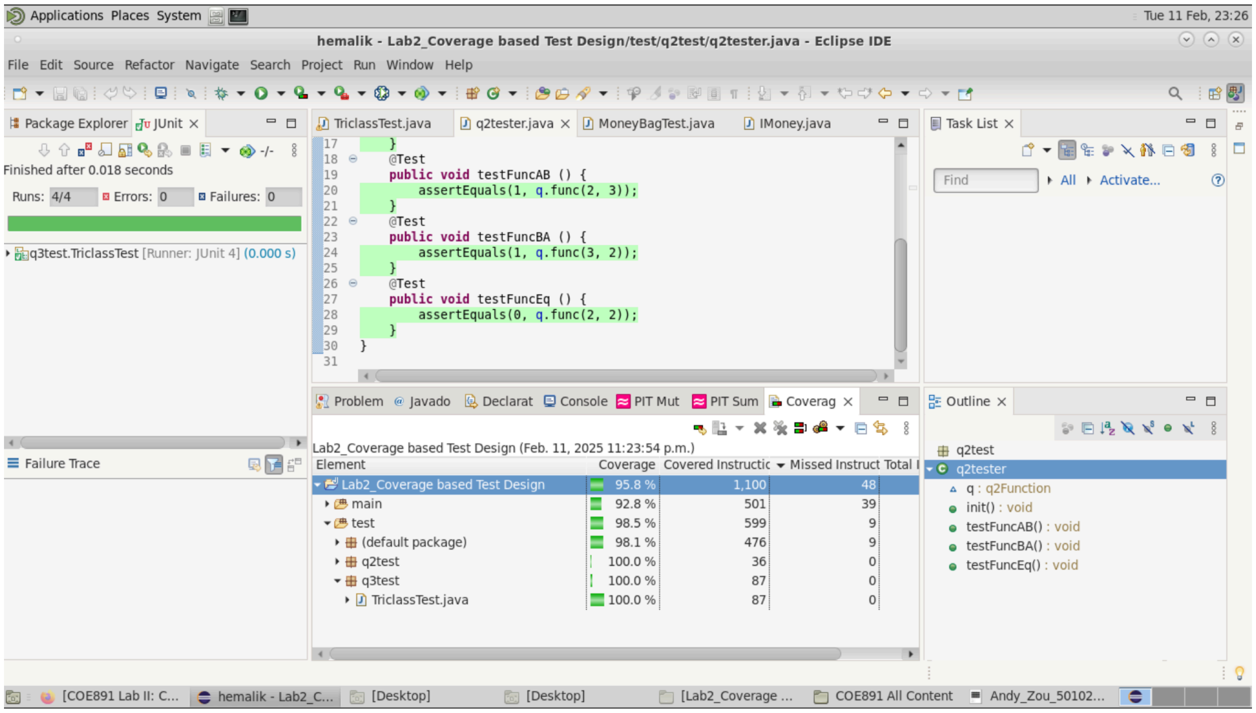
```java
    }
    @Test
    public void testFuncAB () {
        assertEquals(1, q.func(2, 3));
    }
    @Test
    public void testFuncBA () {
        assertEquals(1, q.func(3, 2));
    }
    @Test
    public void testFuncEq () {
        assertEquals(0, q.func(2, 2));
    }
}
```

Lab2_Coverage based Test Design (Feb. 11, 2025 11:23:54 p.m.)

Element	Coverage	Covered Instructic	Missed Instruct	Total I
Lab2_Coverage based Test Design	95.8 %	1,100	48	
main	92.8 %	501	39	
test	98.5 %	599	9	
(default package)	98.1 %	476	9	
q2test	100.0 %	36	0	
q2tester.java	100.0 %	36	0	
q3test	100.0 %	87	0	

q2test
q2tester
q : q2Function
init() : void
testFuncAB() : void
testFuncBA() : void
testFuncEq() : void

Q2: We analyzed the test coverage of our q2test Java program using Clover to evaluate statement and branch coverage. Initially, some execution paths were not fully tested, leading to incomplete coverage. To address this, we introduced additional test cases such as testFuncEq(), ensuring all logical branches were executed. The updated Clover report now confirms 100% statement coverage for q2tester.java, verifying that all conditions in the function have been effectively tested.



Finished after 0.018 seconds

Runs: 4/4 Errors: 0 Failures: 0

q3test.TriclassTest [Runner: JUnit 4] (0.000 s)

```java
    }
    @Test
    public void testFuncAB () {
        assertEquals(1, q.func(2, 3));
    }
    @Test
    public void testFuncBA () {
        assertEquals(1, q.func(3, 2));
    }
    @Test
    public void testFuncEq () {
        assertEquals(0, q.func(2, 2));
    }
}
```

Lab2_Coverage based Test Design (Feb. 11, 2025 11:23:54 p.m.)

Element	Coverage	Covered Instructic	Missed Instruct	Total I
Lab2_Coverage based Test Design	95.8 %	1,100	48	
main	92.8 %	501	39	
test	98.5 %	599	9	
(default package)	98.1 %	476	9	
q2test	100.0 %	36	0	
q3test	100.0 %	87	0	
TriclassTest.java	100.0 %	87	0	

q2test
q2tester
q : q2Function
init() : void
testFuncAB() : void
testFuncBA() : void
testFuncEq() : void

Q3: For the TriclassTest program, we tested different triangle classifications by providing various sets of inputs. The test suite covers cases for equilateral, isosceles, scalene, and invalid triangles while enforcing triangle inequality rules. By fixing two sides at 5 and varying the third, we ensured boundary conditions were met. The Clover report indicates 100% test coverage, confirming that all relevant code statements were executed and validating the correctness of the classification logic