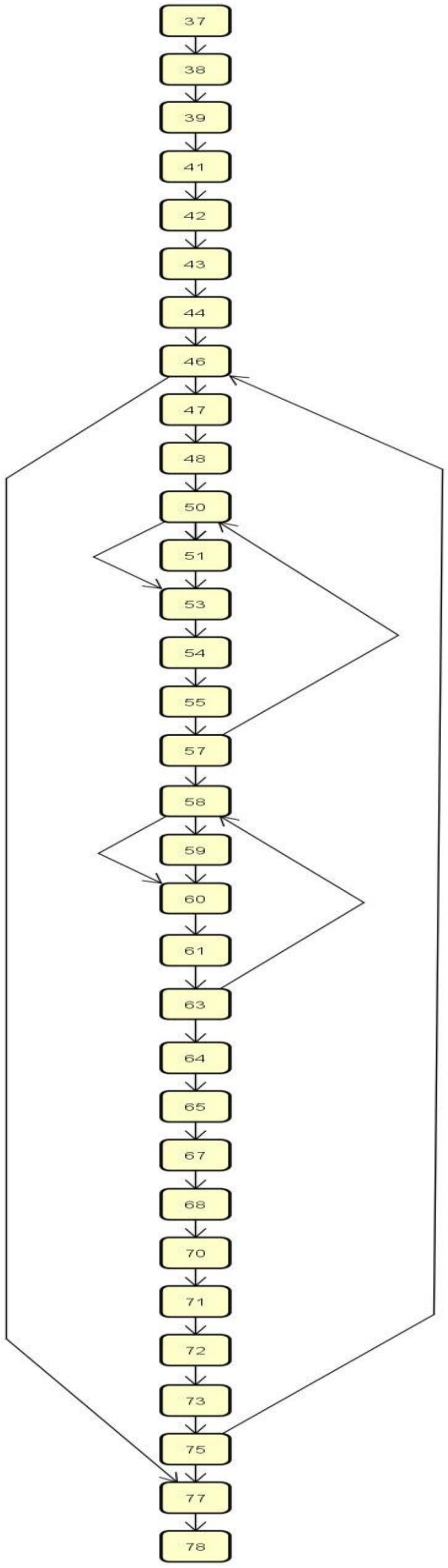
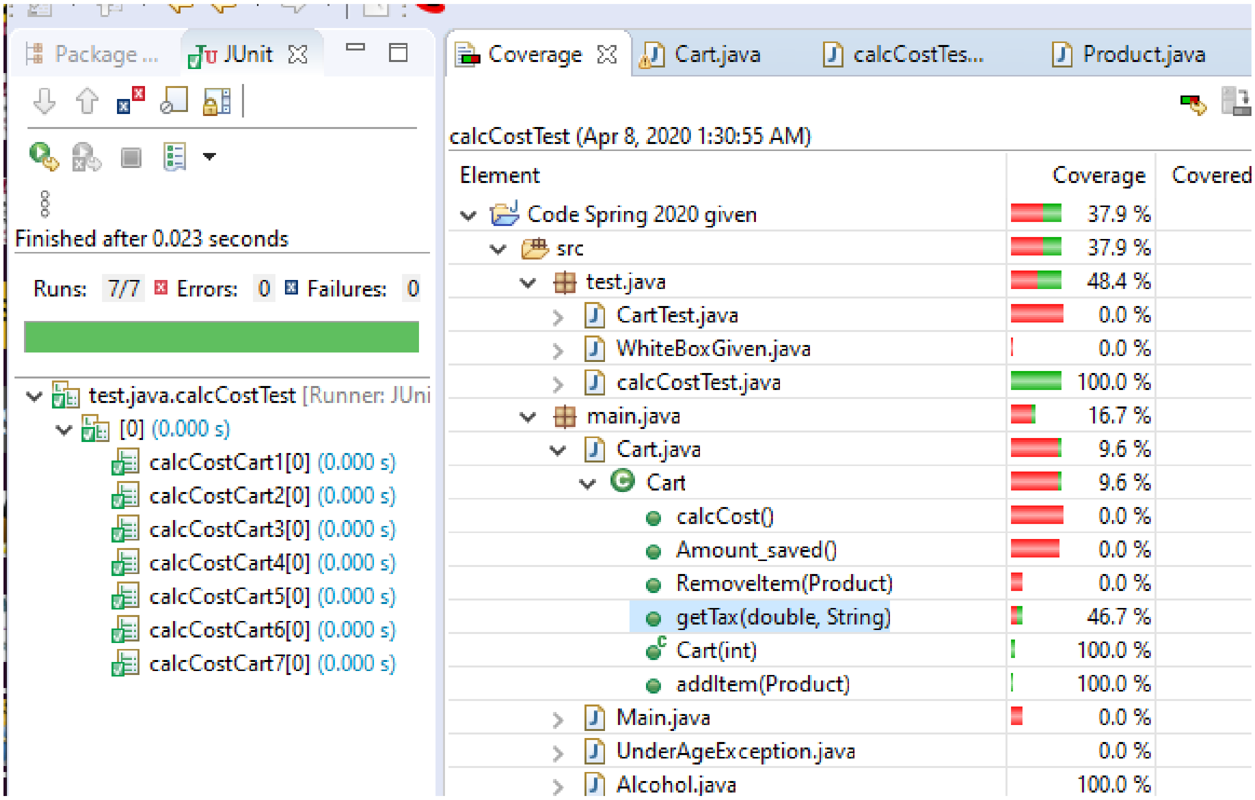
Control Flow Graph for Amount_saved().



The Passing of calcCostTest.java, Including coverage.



Finished after 0.023 seconds

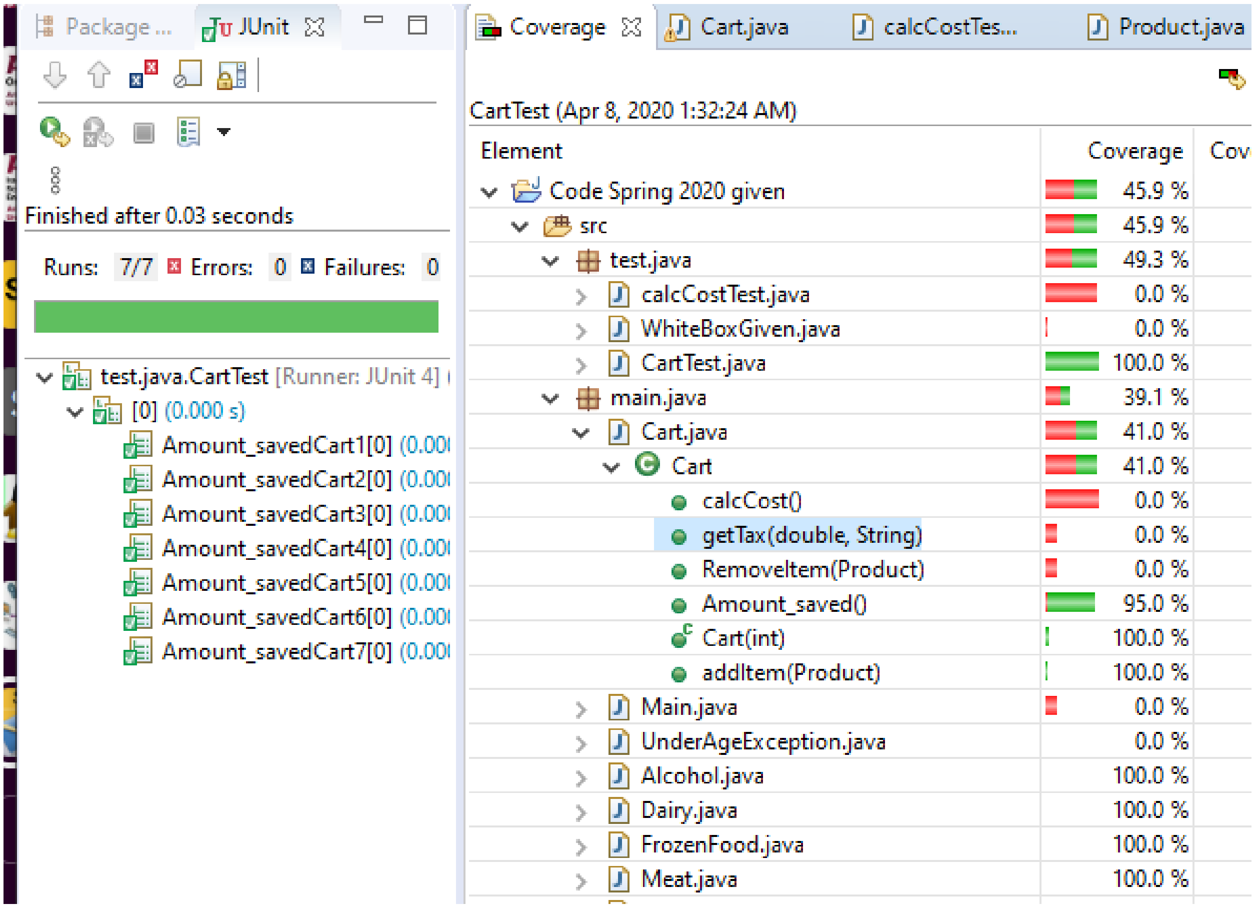
Runs: 7/7 Errors: 0 Failures: 0

- test.java.calcCostTest [Runner: JUni
  - [0] (0.000 s)
    - calcCostCart1[0] (0.000 s)
    - calcCostCart2[0] (0.000 s)
    - calcCostCart3[0] (0.000 s)
    - calcCostCart4[0] (0.000 s)
    - calcCostCart5[0] (0.000 s)
    - calcCostCart6[0] (0.000 s)
    - calcCostCart7[0] (0.000 s)

calcCostTest (Apr 8, 2020 1:30:55 AM)

Element	Coverage	Covered
Code Spring 2020 given	37.9 %	
src	37.9 %	
test.java	48.4 %	
CartTest.java	0.0 %	
WhiteBoxGiven.java	0.0 %	
calcCostTest.java	100.0 %	
main.java	16.7 %	
Cart.java	9.6 %	
Cart	9.6 %	
calcCost()	0.0 %	
Amount_saved()	0.0 %	
RemoveItem(Product)	0.0 %	
getTax(double, String)	46.7 %	
Cart(int)	100.0 %	
addItem(Product)	100.0 %	
Main.java	0.0 %	
UnderAgeException.java	0.0 %	
Alcohol.java	100.0 %	

The Passing of CartTest.java and its coverage.



Finished after 0.03 seconds

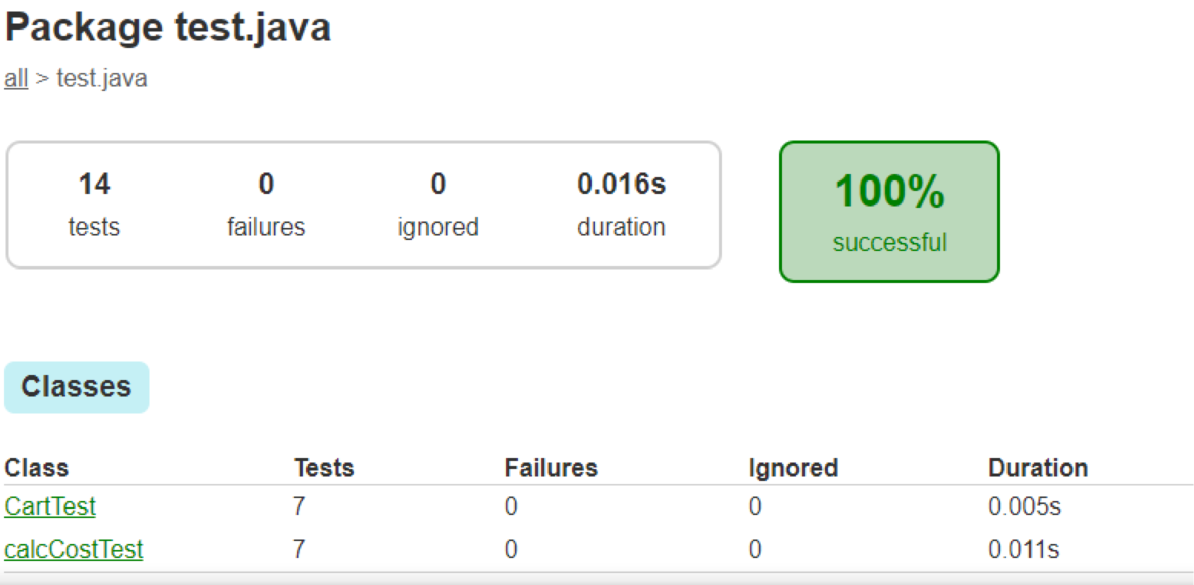
Runs: 7/7 Errors: 0 Failures: 0

```
test.java.CartTest [Runner: JUnit 4] (
  [0] (0.000 s)
    Amount_savedCart1[0] (0.00
    Amount_savedCart2[0] (0.00
    Amount_savedCart3[0] (0.00
    Amount_savedCart4[0] (0.00
    Amount_savedCart5[0] (0.00
    Amount_savedCart6[0] (0.00
    Amount_savedCart7[0] (0.00
```

CartTest (Apr 8, 2020 1:32:24 AM)

Element	Coverage	Cov
Code Spring 2020 given	45.9 %	
src	45.9 %	
test.java	49.3 %	
calcCostTest.java	0.0 %	
WhiteBoxGiven.java	0.0 %	
CartTest.java	100.0 %	
main.java	39.1 %	
Cart.java	41.0 %	
Cart	41.0 %	
calcCost()	0.0 %	
getTax(double, String)	0.0 %	
RemoveItem(Product)	0.0 %	
Amount_saved()	95.0 %	
Cart(int)	100.0 %	
addItem(Product)	100.0 %	
Main.java	0.0 %	
UnderAgeException.java	0.0 %	
Alcohol.java	100.0 %	
Dairy.java	100.0 %	
FrozenFood.java	100.0 %	
Meat.java	100.0 %	

First gradle build.



Package test.java

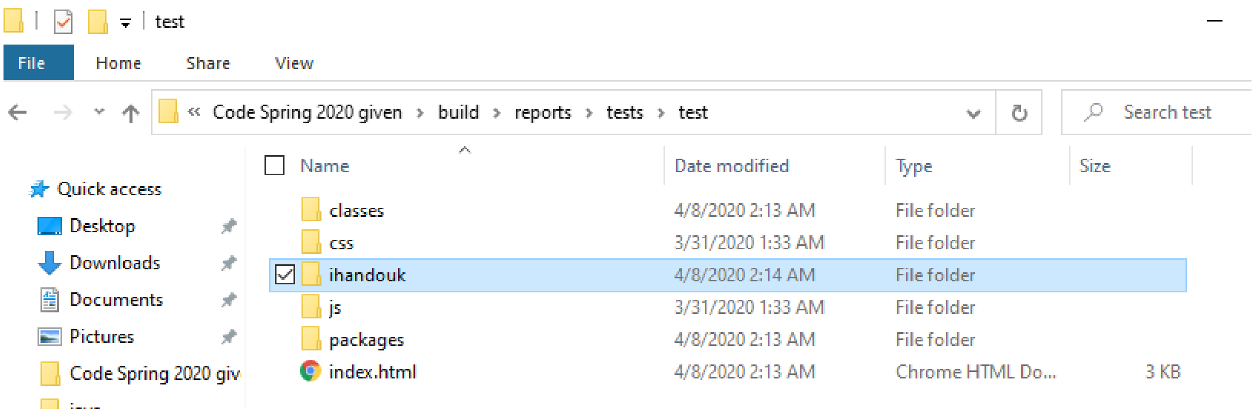
all > test.java

14	0	0	0.016s
tests	failures	ignored	duration

100%
successful

Classes

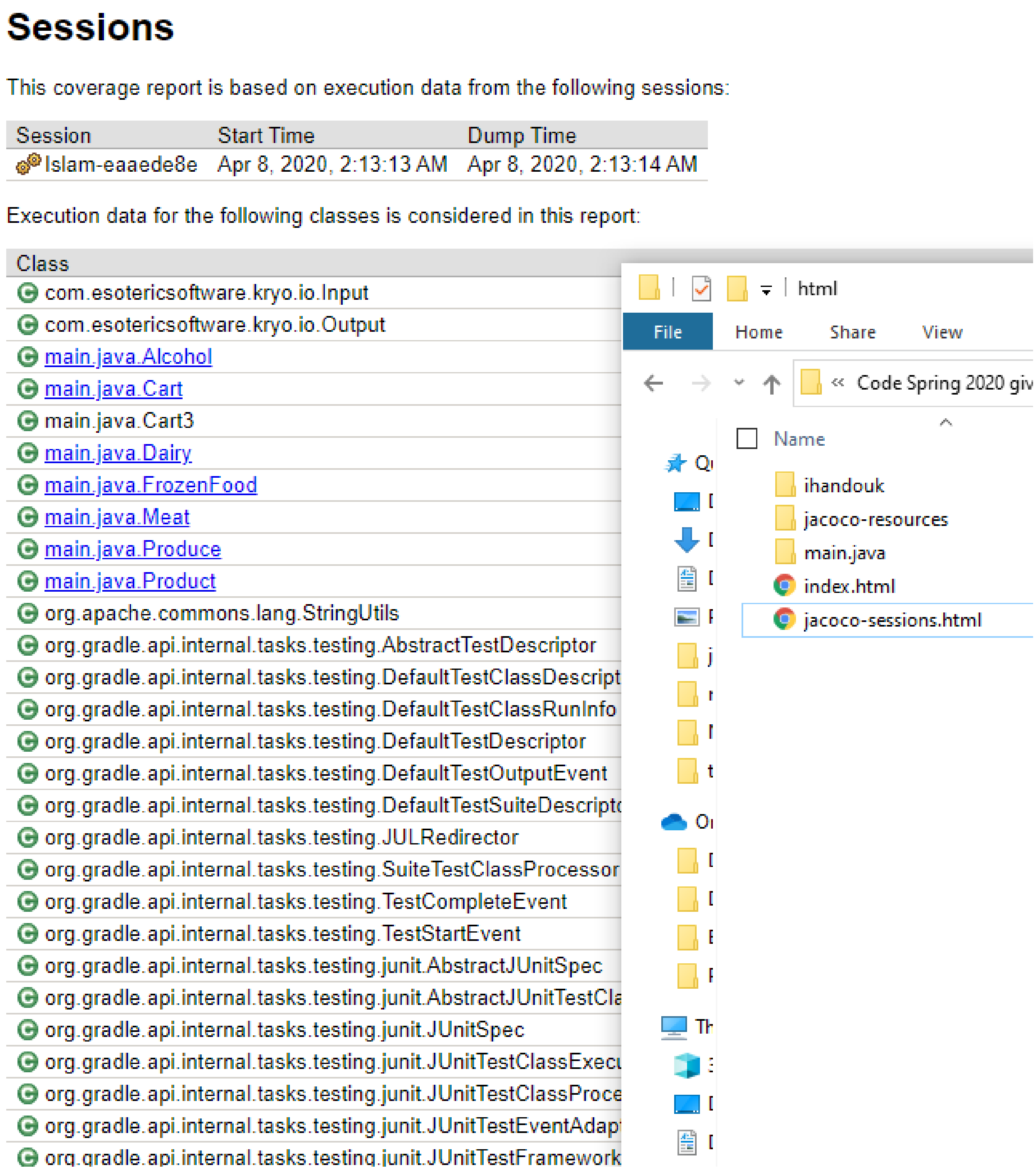
Class	Tests	Failures	Ignored	Duration
CartTest	7	0	0	0.005s
calcCostTest	7	0	0	0.011s



« Code Spring 2020 given › build › reports › tests › test

Name	Date modified	Type	Size
classes	4/8/2020 2:13 AM	File folder	
css	3/31/2020 1:33 AM	File folder	
☑ ihandouk	4/8/2020 2:14 AM	File folder	
js	3/31/2020 1:33 AM	File folder	
packages	4/8/2020 2:13 AM	File folder	
index.html	4/8/2020 2:13 AM	Chrome HTML Do...	3 KB

First Jacoco test report.



Sessions

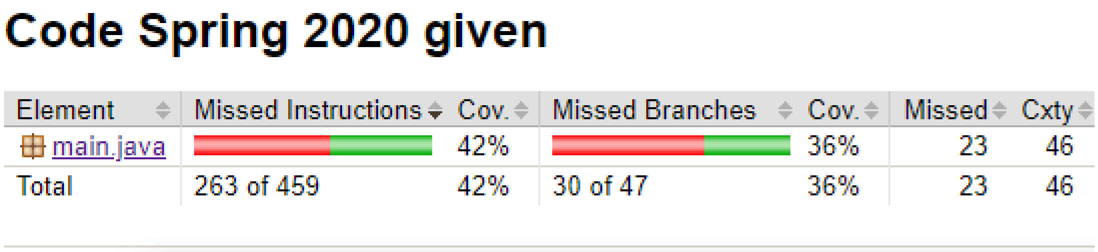
This coverage report is based on execution data from the following sessions:

Session	Start Time	Dump Time
Islam-eaaede8e	Apr 8, 2020, 2:13:13 AM	Apr 8, 2020, 2:13:14 AM

Execution data for the following classes is considered in this report:

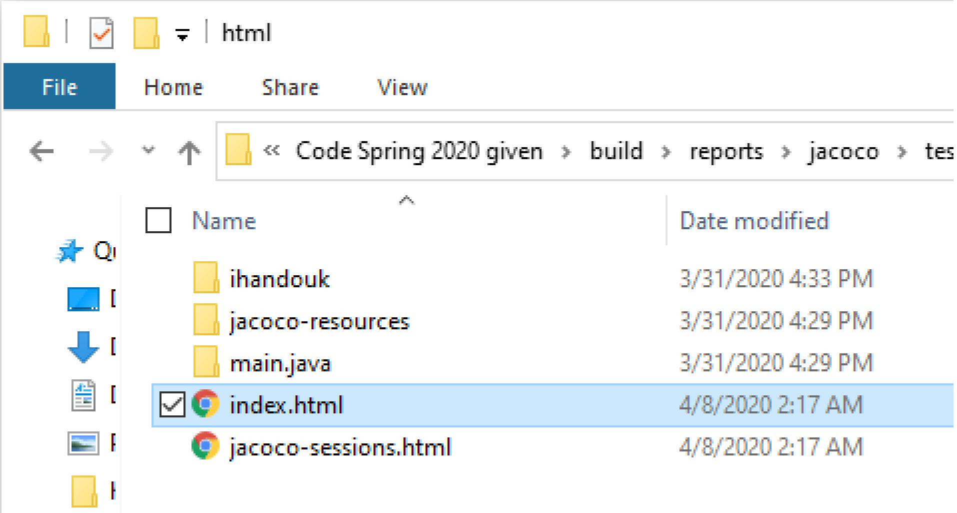
Class
com.esotericsoftware.kryo.io.Input
com.esotericsoftware.kryo.io.Output
main.java.Alcohol
main.java.Cart
main.java.Cart3
main.java.Dairy
main.java.FrozenFood
main.java.Meat
main.java.Produce
main.java.Product
org.apache.commons.lang.StringUtils
org.gradle.api.internal.tasks.testing.AbstractTestDescriptor
org.gradle.api.internal.tasks.testing.DefaultTestClassDescript
org.gradle.api.internal.tasks.testing.DefaultTestClassRunInfo
org.gradle.api.internal.tasks.testing.DefaultTestDescriptor
org.gradle.api.internal.tasks.testing.DefaultTestOutputEvent
org.gradle.api.internal.tasks.testing.DefaultTestSuiteDescripto
org.gradle.api.internal.tasks.testing.JULRedirector
org.gradle.api.internal.tasks.testing.SuiteTestClassProcessor
org.gradle.api.internal.tasks.testing.TestCompleteEvent
org.gradle.api.internal.tasks.testing.TestStartEvent
org.gradle.api.internal.tasks.testing.junit.AbstractJUnitSpec
org.gradle.api.internal.tasks.testing.junit.AbstractJUnitTestCla
org.gradle.api.internal.tasks.testing.junit.JUnitSpec
org.gradle.api.internal.tasks.testing.junit.JUnitTestClassExecu
org.gradle.api.internal.tasks.testing.junit.JUnitTestClassProce
org.gradle.api.internal.tasks.testing.junit.JUnitTestEventAdap
org.gradle.api.internal.tasks.testing.junit.JUnitTestFramework

Jacoco code coverage report.



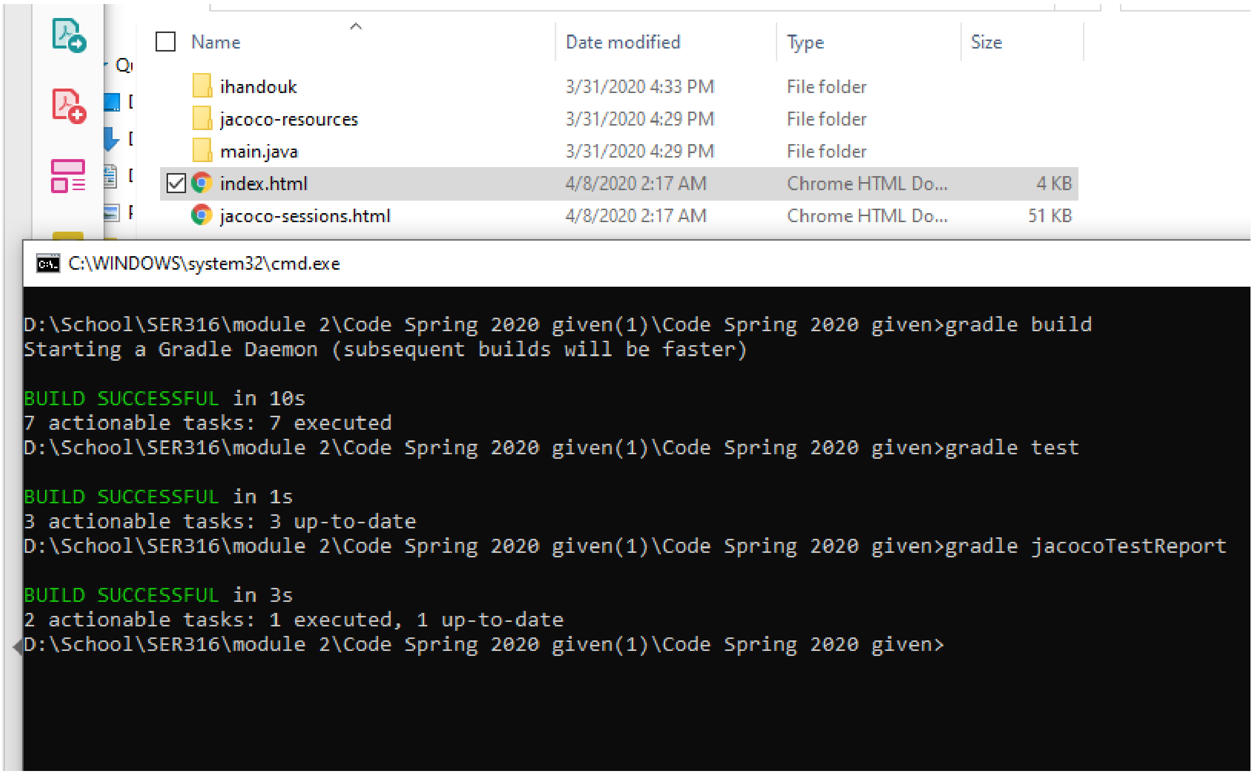
Code Spring 2020 given

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty
main.java	🟥🟩	42%	🟥🟩	36%	23	46
Total	263 of 459	42%	30 of 47	36%	23	46



Name	Date modified
ihandouk	3/31/2020 4:33 PM
jacoco-resources	3/31/2020 4:29 PM
main.java	3/31/2020 4:29 PM
☑ index.html	4/8/2020 2:17 AM
jacoco-sessions.html	4/8/2020 2:17 AM

Command window, running gradle build, gradle test, and gradle jacoco.

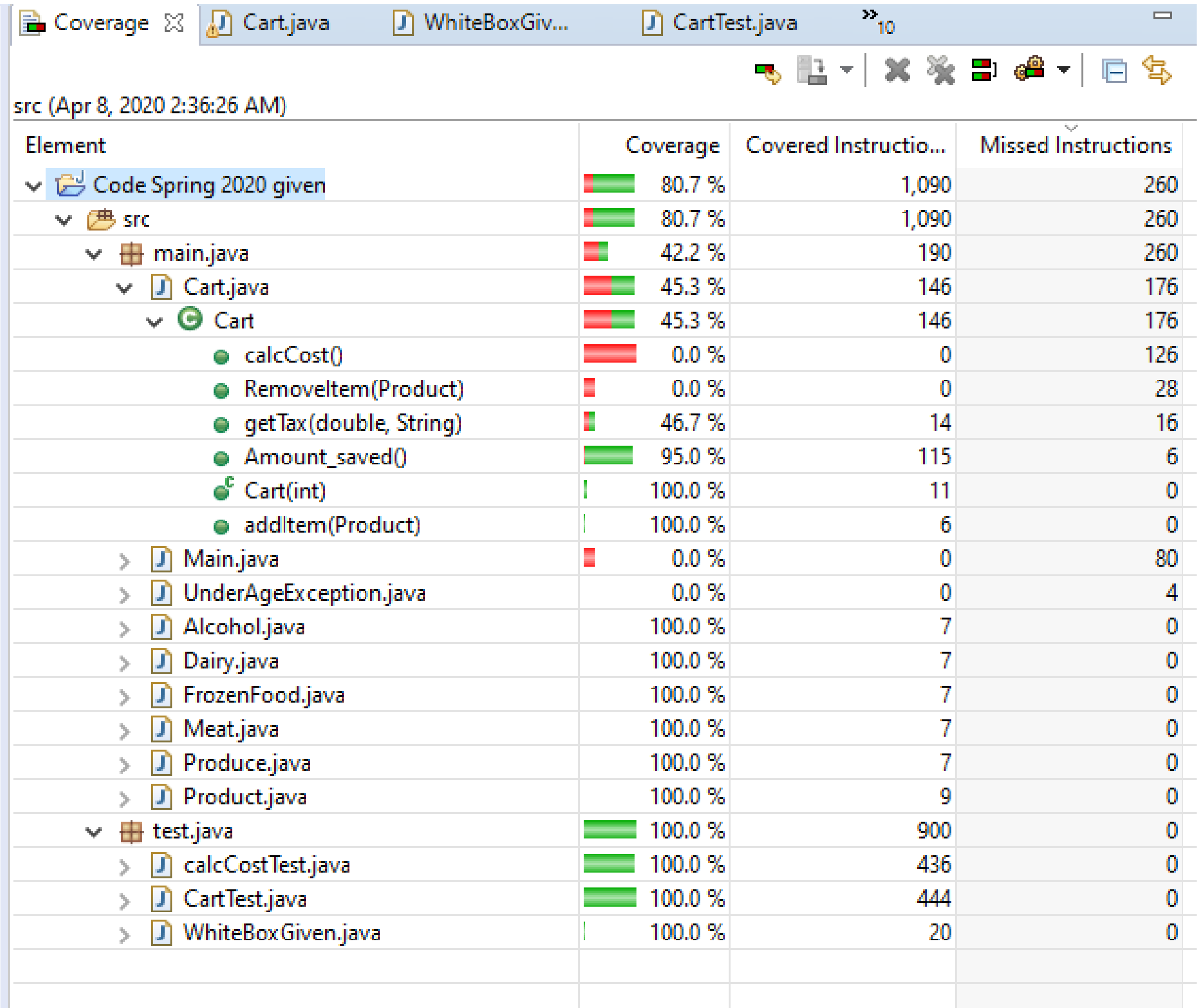


Name	Date modified	Type	Size
ihandouk	3/31/2020 4:33 PM	File folder	
jacoco-resources	3/31/2020 4:29 PM	File folder	
main.java	3/31/2020 4:29 PM	File folder	
☑ index.html	4/8/2020 2:17 AM	Chrome HTML Do...	4 KB
jacoco-sessions.html	4/8/2020 2:17 AM	Chrome HTML Do...	51 KB

```
D:\School\SER316\module 2\Code Spring 2020 given(1)\Code Spring 2020 given>gradle build
Starting a Gradle Daemon (subsequent builds will be faster)

BUILD SUCCESSFUL in 10s
7 actionable tasks: 7 executed
D:\School\SER316\module 2\Code Spring 2020 given(1)\Code Spring 2020 given>gradle test

BUILD SUCCESSFUL in 1s
3 actionable tasks: 3 up-to-date
D:\School\SER316\module 2\Code Spring 2020 given(1)\Code Spring 2020 given>gradle jacocoTestReport

BUILD SUCCESSFUL in 3s
2 actionable tasks: 1 executed, 1 up-to-date
D:\School\SER316\module 2\Code Spring 2020 given(1)\Code Spring 2020 given>
```

Questions:

1. What is the overall code coverage for the whole program?
   a. The overall coverage is for the whole program is 45.9%.
2. What is the code coverage for Cart.java with your tests from CartTest.java?
   a. The coverage for Cart.java is 41%, while the coverage for CartTest.java is 100%. The total average is 70.5% coverage.
3. What is the Code Coverage you were able to reach for the class as a whole?
   a. The code coverage for the class is 80.7%



src (Apr 8, 2020 2:36:26 AM)

Element	Coverage	Covered Instructio...	Missed Instructions
Code Spring 2020 given	80.7 %	1,090	260
src	80.7 %	1,090	260
main.java	42.2 %	190	260
Cart.java	45.3 %	146	176
Cart	45.3 %	146	176
calcCost()	0.0 %	0	126
Removeltem(Product)	0.0 %	0	28
getTax(double, String)	46.7 %	14	16
Amount_saved()	95.0 %	115	6
Cart(int)	100.0 %	11	0
addItem(Product)	100.0 %	6	0
Main.java	0.0 %	0	80
UnderAgeException.java	0.0 %	0	4
Alcohol.java	100.0 %	7	0
Dairy.java	100.0 %	7	0
FrozenFood.java	100.0 %	7	0
Meat.java	100.0 %	7	0
Produce.java	100.0 %	7	0
Product.java	100.0 %	9	0
test.java	100.0 %	900	0
calcCostTest.java	100.0 %	436	0
CartTest.java	100.0 %	444	0
WhiteBoxGiven.java	100.0 %	20	0