Report 0

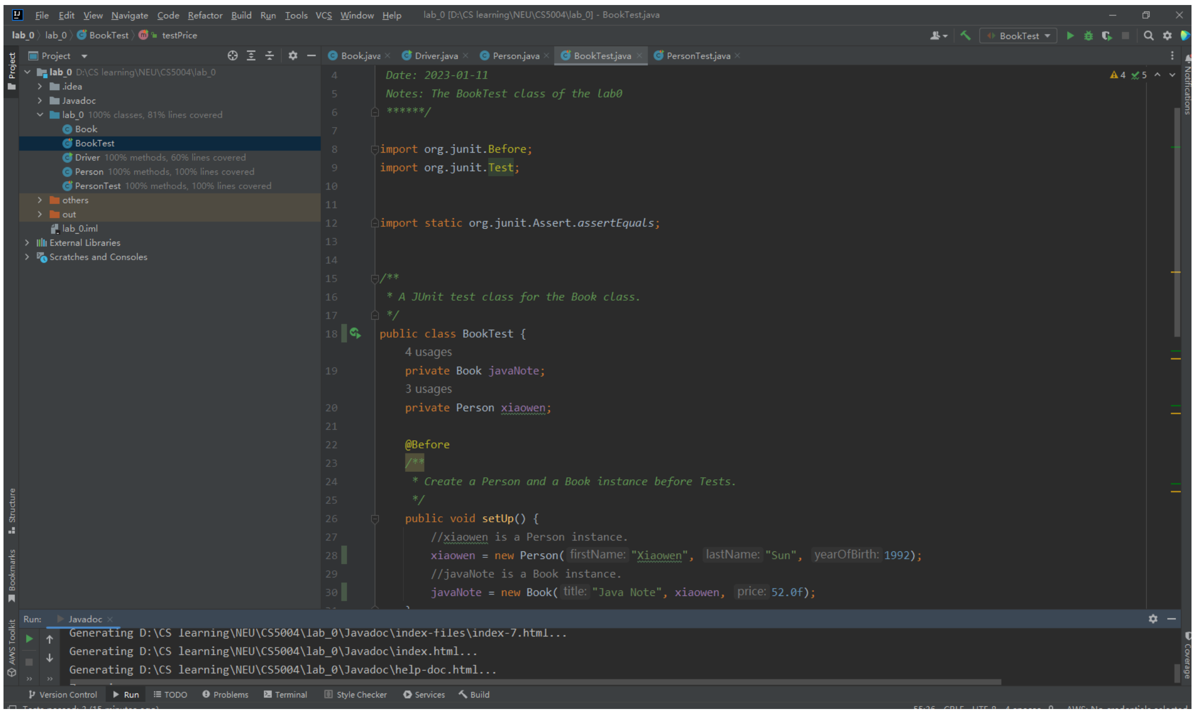
1. Reflection

In this assessment, I learned to establish the Java coding environment, IDE installation, as well as running Java programs by IDE, command line and online compiler.
I learned to create class, instance and write a driver in a Java project. In terms of test, I learned to write simple unit tests for the program and utilizing a few modules in junit.

2. Required Task Elements

Part 5: Running an Existing Project and Running JUnit Tests

IntelliJ project screen shot

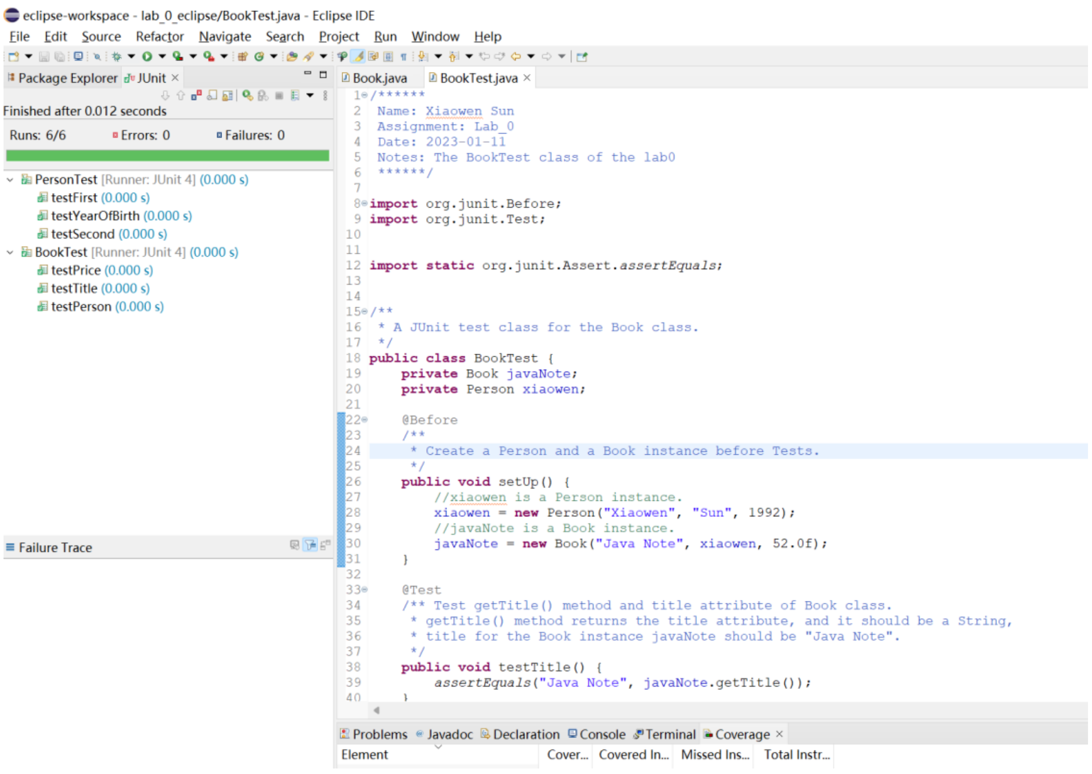

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
lab_0 [D:\CS learning\NEU\CS5004\lab_0] - BookTest.java
lab_0 〉 lab_0 〉 BookTest 〉 testPrice
Project
Book.java Driver.java Person.java BookTest.java PersonTest.java
Version Control Run TODO Problems Terminal Style Checker Services Build

```
lab_0 D:\CS learning\NEU\CS5004\lab_0
  .idea
  Javadoc
  lab_0 100% classes, 81% lines covered
    Book
    BookTest
    Driver 100% methods, 60% lines covered
    Person 100% methods, 100% lines covered
    PersonTest 100% methods, 100% lines covered
  others
  out
  lab_0.iml
External Libraries
Scratches and Consoles
```

```java
    Date: 2023-01-11
    Notes: The BookTest class of the lab0
    ******/

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


import static org.junit.Assert.assertEquals;


/**
 * A JUnit test class for the Book class.
 */
public class BookTest {
    4 usages
    private Book javaNote;
    3 usages
    private Person xiaowen;

    @Before
    /**
     * Create a Person and a Book instance before Tests.
     */
    public void setUp() {
        //xiaowen is a Person instance.
        xiaowen = new Person( firstName: "Xiaowen",  lastName: "Sun",  yearOfBirth: 1992);
        //javaNote is a Book instance.
        javaNote = new Book( title: "Java Note", xiaowen,  price: 52.0f);
```

```
Run:   Javadoc
Generating D:\CS learning\NEU\CS5004\lab_0\Javadoc\index-files\index-7.html...
Generating D:\CS learning\NEU\CS5004\lab_0\Javadoc\index.html...
Generating D:\CS learning\NEU\CS5004\lab_0\Javadoc\help-doc.html...
```

Eclipse project screen shot

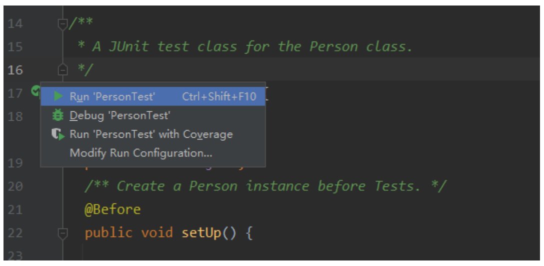


eclipse-workspace - lab_0_eclipse/BookTest.java - Eclipse IDE

```
Finished after 0.012 seconds
Runs: 6/6        Errors: 0        Failures: 0

PersonTest [Runner: JUnit 4] (0.000 s)
    testFirst (0.000 s)
    testYearOfBirth (0.000 s)
    testSecond (0.000 s)
BookTest [Runner: JUnit 4] (0.000 s)
    testPrice (0.000 s)
    testTitle (0.000 s)
    testPerson (0.000 s)
```

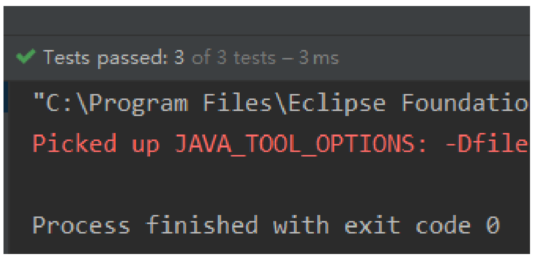
```java
/******
 Name: Xiaowen Sun
 Assignment: Lab_0
 Date: 2023-01-11
 Notes: The BookTest class of the lab0
 ******/

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


import static org.junit.Assert.assertEquals;


/**
 * A JUnit test class for the Book class.
 */
public class BookTest {
    private Book javaNote;
    private Person xiaowen;

    @Before
    /**
     * Create a Person and a Book instance before Tests.
     */
    public void setUp() {
        //xiaowen is a Person instance.
        xiaowen = new Person("Xiaowen", "Sun", 1992);
        //javaNote is a Book instance.
        javaNote = new Book("Java Note", xiaowen, 52.0f);
    }

    @Test
    /** Test getTitle() method and title attribute of Book class.
     * getTitle() method returns the title attribute, and it should be a String,
     * title for the Book instance javaNote should be "Java Note".
     */
    public void testTitle() {
        assertEquals("Java Note", javaNote.getTitle());
    }
```

Test for person class, 3 passed



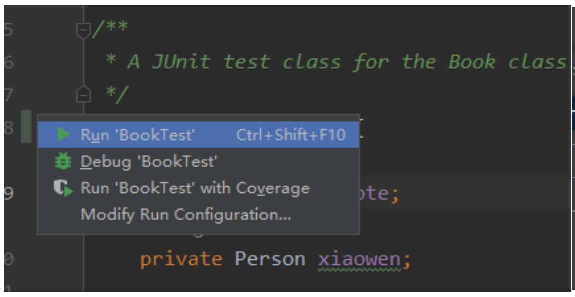
```java
/**
 * A JUnit test class for the Person class.
 */

/** Create a Person instance before Tests. */
@Before
public void setUp() {
```

Run 'PersonTest'   Ctrl+Shift+F10
Debug 'PersonTest'
Run 'PersonTest' with Coverage
Modify Run Configuration...



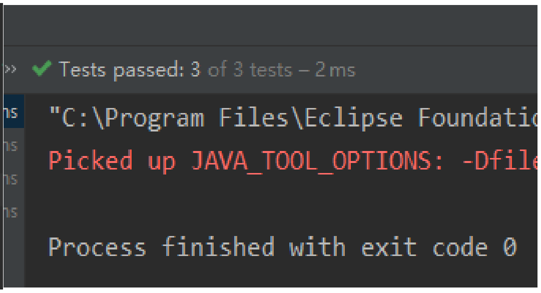
```
Tests passed: 3 of 3 tests – 3 ms

"C:\Program Files\Eclipse Foundatio
Picked up JAVA_TOOL_OPTIONS: -Dfile

Process finished with exit code 0
```

Test for book class, 3 passed



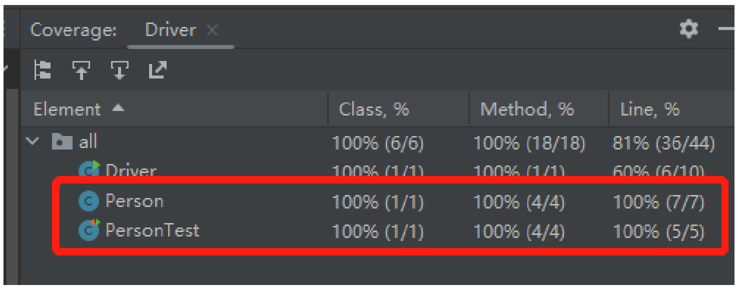
```java
/**
 * A JUnit test class for the Book class
 */

private Person xiaowen;
```

Run 'BookTest'   Ctrl+Shift+F10
Debug 'BookTest'
Run 'BookTest' with Coverage
Modify Run Configuration...



```
Tests passed: 3 of 3 tests – 2 ms

"C:\Program Files\Eclipse Foundatio
Picked up JAVA_TOOL_OPTIONS: -Dfile

Process finished with exit code 0
```

The coverage of booktest and persontest is 100%

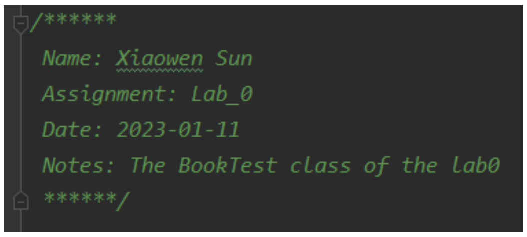


Coverage: Driver

Element	Class, %	Method, %	Line, %
all	100% (6/6)	100% (18/18)	81% (36/44)
Driver	100% (1/1)	100% (1/1)	60% (6/10)
Person	100% (1/1)	100% (4/4)	100% (7/7)
PersonTest	100% (1/1)	100% (4/4)	100% (5/5)

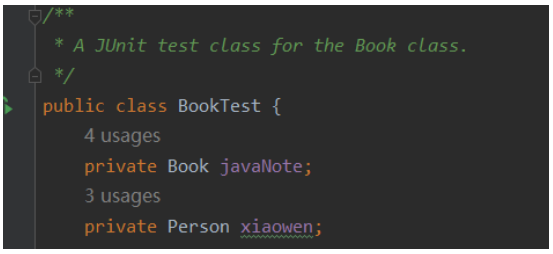
Part 6: Documenting Your Code

Top comment block



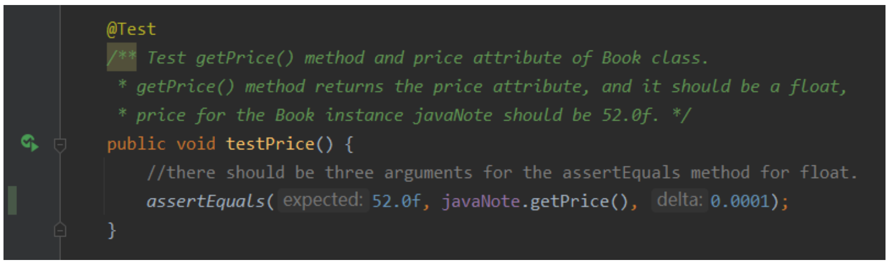
```
/******
 Name: Xiaowen Sun
 Assignment: Lab_0
 Date: 2023-01-11
 Notes: The BookTest class of the lab0
 ******/
```

Class comment block



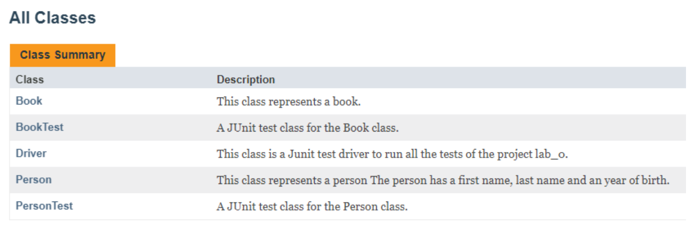
```
/**
 * A JUnit test class for the Book class.
 */
public class BookTest {
    4 usages
    private Book javaNote;
    3 usages
    private Person xiaowen;
```

Function comment block and Java doc line



```
@Test
/** Test getPrice() method and price attribute of Book class.
 * getPrice() method returns the price attribute, and it should be a float,
 * price for the Book instance javaNote should be 52.0f. */
public void testPrice() {
    //there should be three arguments for the assertEquals method for float.
    assertEquals( expected: 52.0f, javaNote.getPrice(),  delta: 0.0001);
}
```

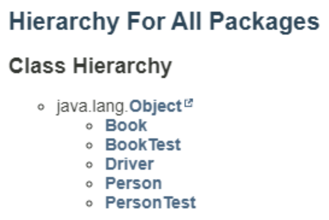
Generated Java Doc



All Classes

Class Summary

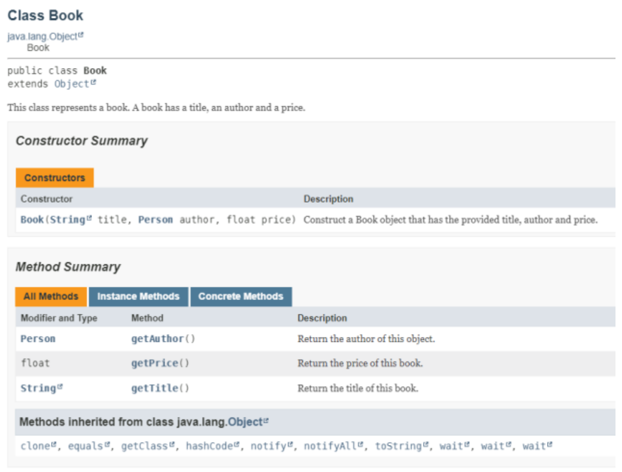
Class	Description
Book	This class represents a book.
BookTest	A JUnit test class for the Book class.
Driver	This class is a Junit test driver to run all the tests of the project lab_o.
Person	This class represents a person The person has a first name, last name and an year of birth.
PersonTest	A JUnit test class for the Person class.



Hierarchy For All Packages

Class Hierarchy

- java.lang.Object
  - Book
  - BookTest
  - Driver
  - Person
  - PersonTest



Class Book

java.lang.Object
    Book

```
public class Book
extends Object
```

This class represents a book. A book has a title, an author and a price.

Constructor Summary

Constructors

Constructor	Description
Book(String title, Person author, float price)	Construct a Book object that has the provided title, author and price.

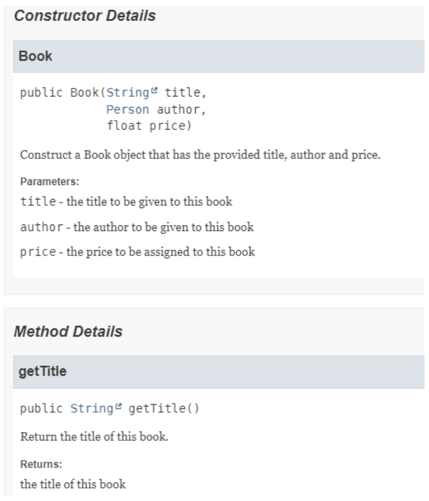
Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
Person	getAuthor()	Return the author of this object.
float	getPrice()	Return the price of this book.
String	getTitle()	Return the title of this book.

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



Constructor Details

Book

```
public Book(String title,
            Person author,
            float price)
```

Construct a Book object that has the provided title, author and price.

Parameters:

title - the title to be given to this book

author - the author to be given to this book

price - the price to be assigned to this book

Method Details

getTitle

```
public String getTitle()
```

Return the title of this book.

Returns:

the title of this book

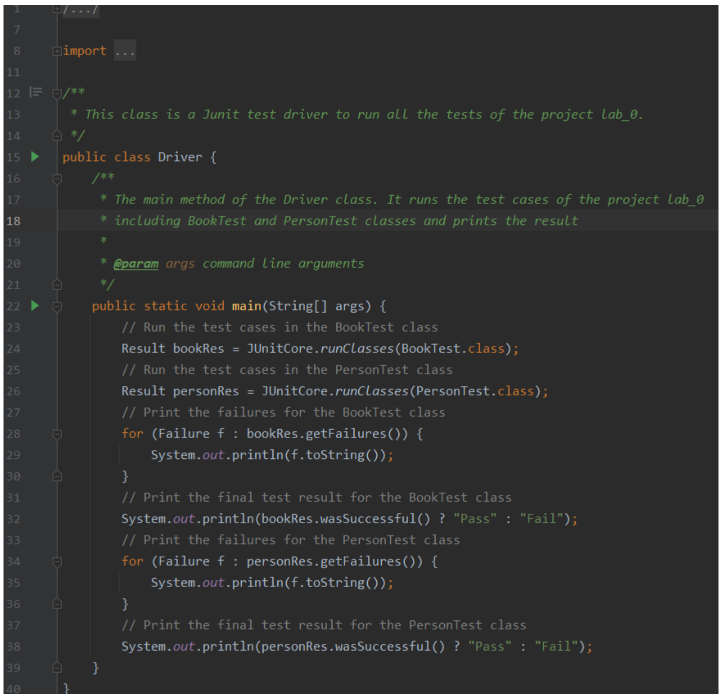
3. Extensions

I tried several extensions as follows.

- Provide screenshots of both Eclipse and IntelliJ, see “2.Required Task Elements”
- Created a driver for this application.

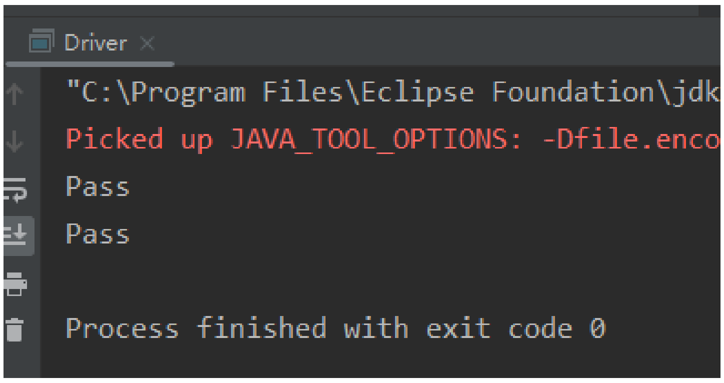
  I created a driver.java, with several modules in junit to run all the tests and print messages.

  Driver code:



```java
/.../

import ...

/**
 * This class is a Junit test driver to run all the tests of the project lab_0.
 */
public class Driver {
    /**
     * The main method of the Driver class. It runs the test cases of the project lab_0
     * including BookTest and PersonTest classes and prints the result
     *
     * @param args command line arguments
     */
    public static void main(String[] args) {
        // Run the test cases in the BookTest class
        Result bookRes = JUnitCore.runClasses(BookTest.class);
        // Run the test cases in the PersonTest class
        Result personRes = JUnitCore.runClasses(PersonTest.class);
        // Print the failures for the BookTest class
        for (Failure f : bookRes.getFailures()) {
            System.out.println(f.toString());
        }
        // Print the final test result for the BookTest class
        System.out.println(bookRes.wasSuccessful() ? "Pass" : "Fail");
        // Print the failures for the PersonTest class
        for (Failure f : personRes.getFailures()) {
            System.out.println(f.toString());
        }
        // Print the final test result for the PersonTest class
        System.out.println(personRes.wasSuccessful() ? "Pass" : "Fail");
    }
}
```

  Driver output:



```
"C:\Program Files\Eclipse Foundation\jdk
Picked up JAVA_TOOL_OPTIONS: -Dfile.enco
Pass
Pass

Process finished with exit code 0
```

- Submitted code as a GitHub link instead

  Github link:

  https://github.com/Shawnsuun/CS5004_PTL/tree/main/lab_0
- High quality submission

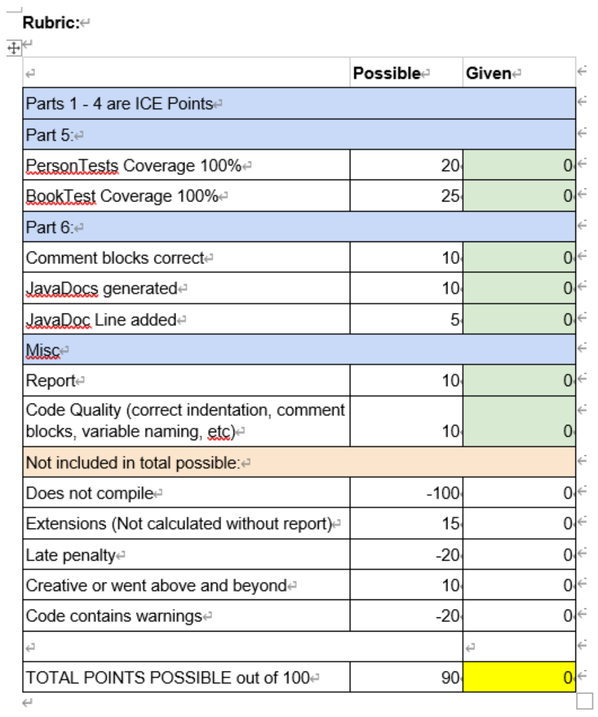
Grading Statement

Part5: The coverage of two tests are both 100%.

Part6: I added Comment blocks correctly in each file, I also wrote java docs for each class ,method, and somewhere necessary in the code, then generated the html formatted javadocs.
Misc: Report is included and the code quality meets the requirement of the lab.
Extensions: I made extensions including:

using Eclipse and IntelliJ
created a driver
submit as a GitHub link
high quality submission

For the above reasons, I think this work deserves a full mark.



Rubric:

	Possible	Given
Parts 1 - 4 are ICE Points		
Part 5:		
PersonTests Coverage 100%	20	0
BookTest Coverage 100%	25	0
Part 6:		
Comment blocks correct	10	0
JavaDocs generated	10	0
JavaDoc Line added	5	0
Misc		
Report	10	0
Code Quality (correct indentation, comment blocks, variable naming, etc)	10	0
Not included in total possible:		
Does not compile	-100	0
Extensions (Not calculated without report)	15	0
Late penalty	-20	0
Creative or went above and beyond	10	0
Code contains warnings	-20	0
TOTAL POINTS POSSIBLE out of 100	90	0