CSE 5321 SOFTWARE TESTING

FALL 2015

HOMEWORK 3

Submitted by: Sneha Suhas Chitre

Student Id: 1001244953

**Problem 1**

Given problem can be formulated as following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sales Volume | Commission Rate | Fee | Status | Gift Card |
| -$0.01 | 0 | -$500.00 | No Sales | No |
| $0.00 | 2.00% | $0.00 | Ordiary Sales | No |
| $24,999.99 | 2.00% | $0.00 | Ordiary Sales | No |
| $25,000.00 | 2.75% | $0.00 | Ordiary Sales | No |
| $99,999.99 | 2.75% | $0.00 | Ordiary Sales | No |
| $100,000.00 | 3.50% | $0.00 | Ordiary Sales | No |
| $200,000.00 | 3.50% | $0.00 | Ordiary Sales | No |
| $200,000.01 | 3.75% | $0.00 | Ordiary Sales | No |
| $400,000.00 | 3.75% | $0.00 | Ordiary Sales | No |
| $400,000.01 | 4.00% | $0.00 | Top Seller Award | Yes |

**Test Case Table** for the above problem is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Input | Expected Output | | |
| Test Case | Sales Volume | Amount Paid | Gift Card | Status |
| 1 | -$0.01 | -$500.00 | No | No Sales |
| 2 | $0.00 | $0.00 | No | Ordiary Sales |
| 3 | $24,999.99 | $499.99 | No | Ordiary Sales |
| 4 | $25,000.00 | $687.50 | No | Ordiary Sales |
| 5 | $99,999.99 | $2,749.99 | No | Ordiary Sales |
| 6 | $100,000.00 | $3,500.00 | No | Ordiary Sales |
| 7 | $200,000.00 | $7,000.00 | No | Ordiary Sales |
| 8 | $200,000.01 | $7,500.00 | No | Ordiary Sales |
| 9 | $400,000.00 | $15,000.00 | No | Ordiary Sales |
| 10 | $400,000.01 | $16,000.00 | Yes | Top Seller Award |

**Decision Table** for above problem is as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Conditions/ Rules** | **1** | **2** | **3** | **4** | **5** | **6** |
| Sales Volume < $0 | Y | N | N | N | N | N |
| $0.00 <= Sales Volume <=24999.99 | N | Y | N | N | N | N |
| $25000.00 <= Sales Volume <=99999.99 | N | N | Y | N | N | N |
| $100000.00 <= Sales Volume <=200000.00 | N | N | N | Y | N | N |
| $200000.01 <= Sales Volume <=400000.00 | N | N | N | N | Y | N |
| Sales Volume > $400000.00 | N | N | N | N | N | Y |
| **Actions** |  |  |  |  |  |  |
| Comission Rate : 0.00% | X |  |  |  |  |  |
| Comission Rate : 2.00% |  | X |  |  |  |  |
| Comission Rate : 2.75% |  |  | X |  |  |  |
| Comission Rate : 3.50% |  |  |  | X |  |  |
| Comission Rate : 3.75% |  |  |  |  | X |  |
| Comission Rate : 4.00% |  |  |  |  |  | X |
| Fee $500 | X |  |  |  |  |  |
| No Fee |  | X | X | X | X | X |
| No Sales | X |  |  |  |  |  |
| Ordinary Sales |  | X | X | X | X |  |
| Top Seller Award |  |  |  |  |  | X |
| Gift Card: Yes |  |  |  |  |  | X |
| Gift Card: No | X | X | X | X | X |  |

**JAVA Code** for above problem is given below, as well as src attached.

**Sales\_status.java**

**public** **enum** Sales\_status {***NO\_SALES***,***ORDINARY\_SALES***,***TOP\_SELLER***};

**EmployeeBonus.java**

**import** javax.swing.\*;

**public** **class** EmployeeBonus {

**public** **double** bonus\_to\_be\_paid;

**public** **boolean** gift\_card\_earned;

**public** **double** gift\_card\_amount;

**public** **boolean** fee\_charged;

**public** **double** fee\_amount;

**public** **static** Sales\_status *sales\_stat*;

**public** **static** **void** main(String args[]){

String SalesVolume;

Double SalesVolumeNew;

SalesVolume= JOptionPane.*showInputDialog*("Enter Sales Volume in currency format($):");

// if (SalesVolume.contains("-"))

SalesVolumeNew = Double.*parseDouble*(SalesVolume);

// calculate\_bonus(SalesVolumeNew);

//JOptionPane.showMessageDialog(null, "Amount Paid: ");

}

**public** **void** calculate\_bonus(**double** sales\_volume) {

**if**(sales\_volume < 0 )

{

bonus\_to\_be\_paid = -500.00;

fee\_charged=**true**;

fee\_amount=500.00;

gift\_card\_earned = **false**;

gift\_card\_amount = 0.00;

*sales\_stat*= Sales\_status.***NO\_SALES***;

JOptionPane.*showMessageDialog*(**null**, "Bonus to be Paid: $" +bonus\_to\_be\_paid+"\nSales\_Status: "+*sales\_stat*+"\nFee Charged: "+fee\_charged +"\nFee Amount: $"+fee\_amount+"\nGift Card Earned: "+gift\_card\_earned+"\nGift Card Ampount: $"+gift\_card\_amount);

//return sales\_volume;

}

**else** **if**(sales\_volume >= 0.00 && sales\_volume <=24999.99 )

{

bonus\_to\_be\_paid = (2.00\*sales\_volume)/100;

gift\_card\_earned = **false**;

gift\_card\_amount = 0.00;

*sales\_stat*= Sales\_status.***ORDINARY\_SALES***;

fee\_charged=**false**;

fee\_amount=0.00;

System.***out***.println("Bonus to be Paid: $" + bonus\_to\_be\_paid + "\nSales\_Status: " + *sales\_stat* + "\nFee Charged: " + fee\_charged + "\nFee Amount: $" + fee\_amount + "\nGift Card Earned: " + gift\_card\_earned + "\nGift Card Ampount: $" + gift\_card\_amount);

JOptionPane.*showMessageDialog*(**null**, "Bonus to be Paid: $" +bonus\_to\_be\_paid+"\nSales\_Status: "+*sales\_stat*+"\nFee Charged: "+fee\_charged +"\nFee Amount: $"+fee\_amount+"\nGift Card Earned: "+gift\_card\_earned+"\nGift Card Ampount: $"+gift\_card\_amount);

}

**else** **if**(sales\_volume >= 25000.00 && sales\_volume <=99999.99)

{

bonus\_to\_be\_paid = (2.75\*sales\_volume)/100;

gift\_card\_earned = **false**;

gift\_card\_amount = 0.00;

*sales\_stat*= Sales\_status.***ORDINARY\_SALES***;

fee\_charged=**false**;

fee\_amount=0.00;

JOptionPane.*showMessageDialog*(**null**, "Bonus to be Paid: $" +bonus\_to\_be\_paid+"\nSales\_Status: "+*sales\_stat*+"\nFee Charged: "+fee\_charged +"\nFee Amount: $"+fee\_amount+"\nGift Card Earned: "+gift\_card\_earned+"\nGift Card Ampount: $"+gift\_card\_amount);

}

**else** **if**(sales\_volume >= 100000.00 && sales\_volume <=200000.00)

{

bonus\_to\_be\_paid = (3.50\*sales\_volume)/100;

gift\_card\_earned = **false**;

gift\_card\_amount = 0.00;

*sales\_stat*= Sales\_status.***ORDINARY\_SALES***;

fee\_charged=**false**;

fee\_amount=0.00;

JOptionPane.*showMessageDialog*(**null**, "Bonus to be Paid: $" +bonus\_to\_be\_paid+"\nSales\_Status: "+*sales\_stat*+"\nFee Charged: "+fee\_charged +"\nFee Amount: $"+fee\_amount+"\nGift Card Earned: "+gift\_card\_earned+"\nGift Card Ampount: $"+gift\_card\_amount);

}

**else** **if**(sales\_volume >= 200000.01 && sales\_volume <=400000.00)

{

bonus\_to\_be\_paid = (3.75\*sales\_volume)/100;

gift\_card\_earned = **false**;

gift\_card\_amount = 0.00;

*sales\_stat*= Sales\_status.***ORDINARY\_SALES***;

fee\_charged=**false**;

fee\_amount=0.00;

JOptionPane.*showMessageDialog*(**null**, "Bonus to be Paid: $" +bonus\_to\_be\_paid+"\nSales\_Status: "+*sales\_stat*+"\nFee Charged: "+fee\_charged +"\nFee Amount: $"+fee\_amount+"\nGift Card Earned: "+gift\_card\_earned+"\nGift Card Ampount: $"+gift\_card\_amount);

}

**else** **if**(sales\_volume >= 400000.01)

{

bonus\_to\_be\_paid = (4.00\*sales\_volume)/100;

gift\_card\_earned = **true**;

gift\_card\_amount = 250.00;

*sales\_stat*= Sales\_status.***TOP\_SELLER***;

fee\_charged=**false**;

fee\_amount=0.00;

JOptionPane.*showMessageDialog*(**null**, "Bonus to be Paid: $" +bonus\_to\_be\_paid+"\nSales\_Status: "+*sales\_stat*+"\nFee Charged: "+fee\_charged +"\nFee Amount: $"+fee\_amount+"\nGift Card Earned: "+gift\_card\_earned+"\nGift Card Ampount: $"+gift\_card\_amount);

}

}

**public** **boolean** isGift\_card\_earned(){

**return** gift\_card\_earned;

}

**public** **double** getBonus\_to\_be\_paid(){

**return** bonus\_to\_be\_paid;

}

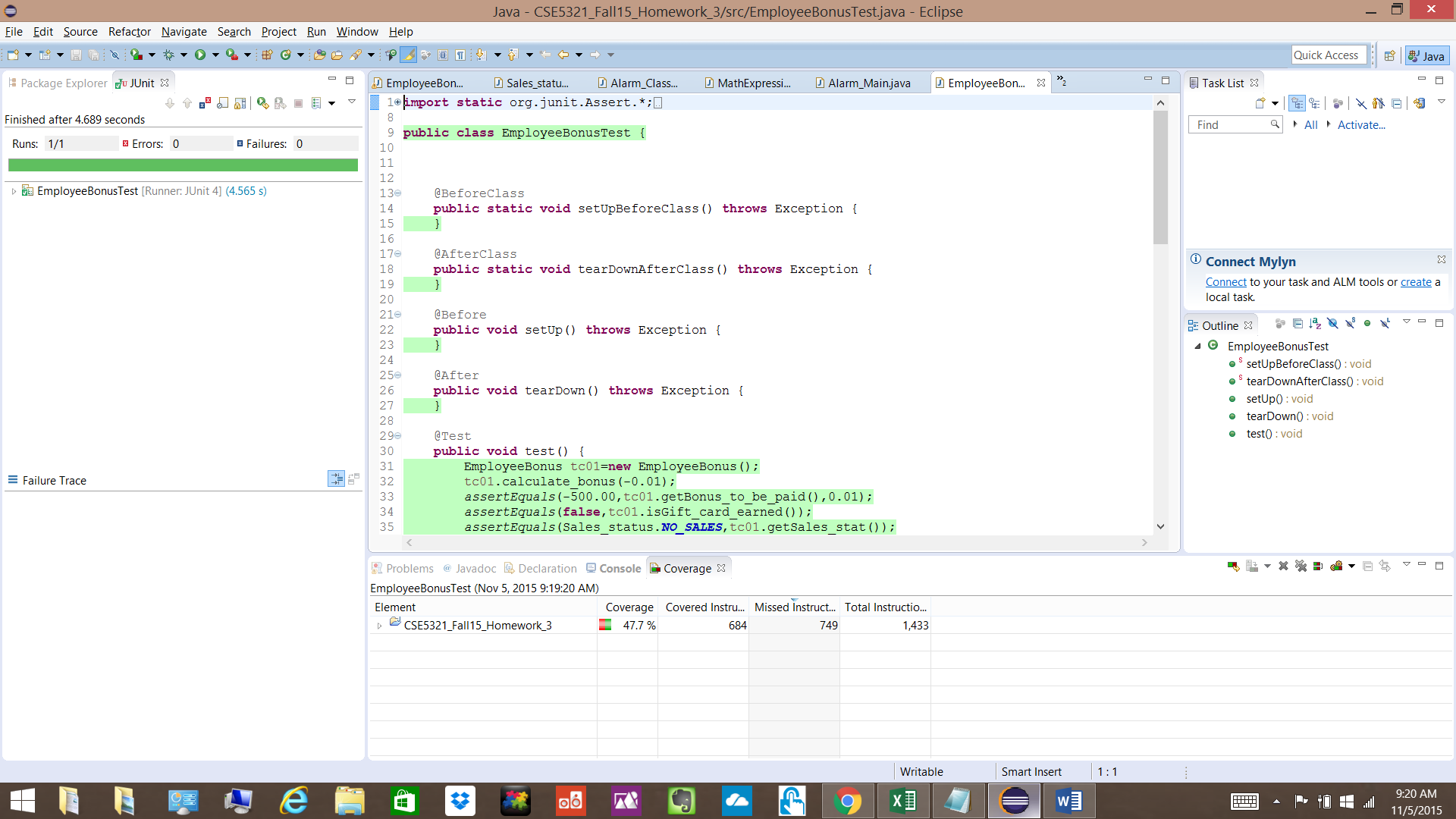
**public** Sales\_status getSales\_stat(){

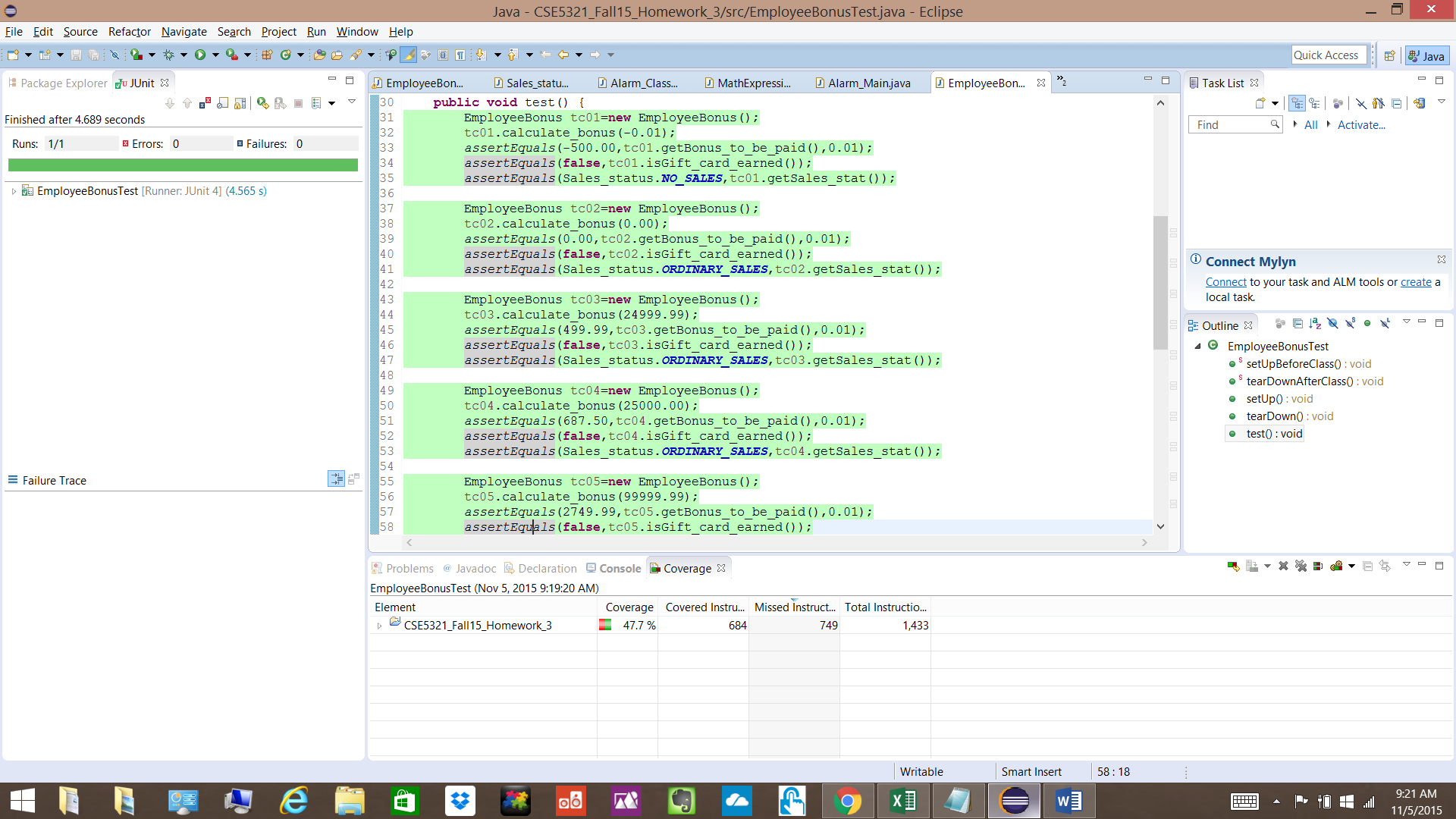
**return** *sales\_stat*;

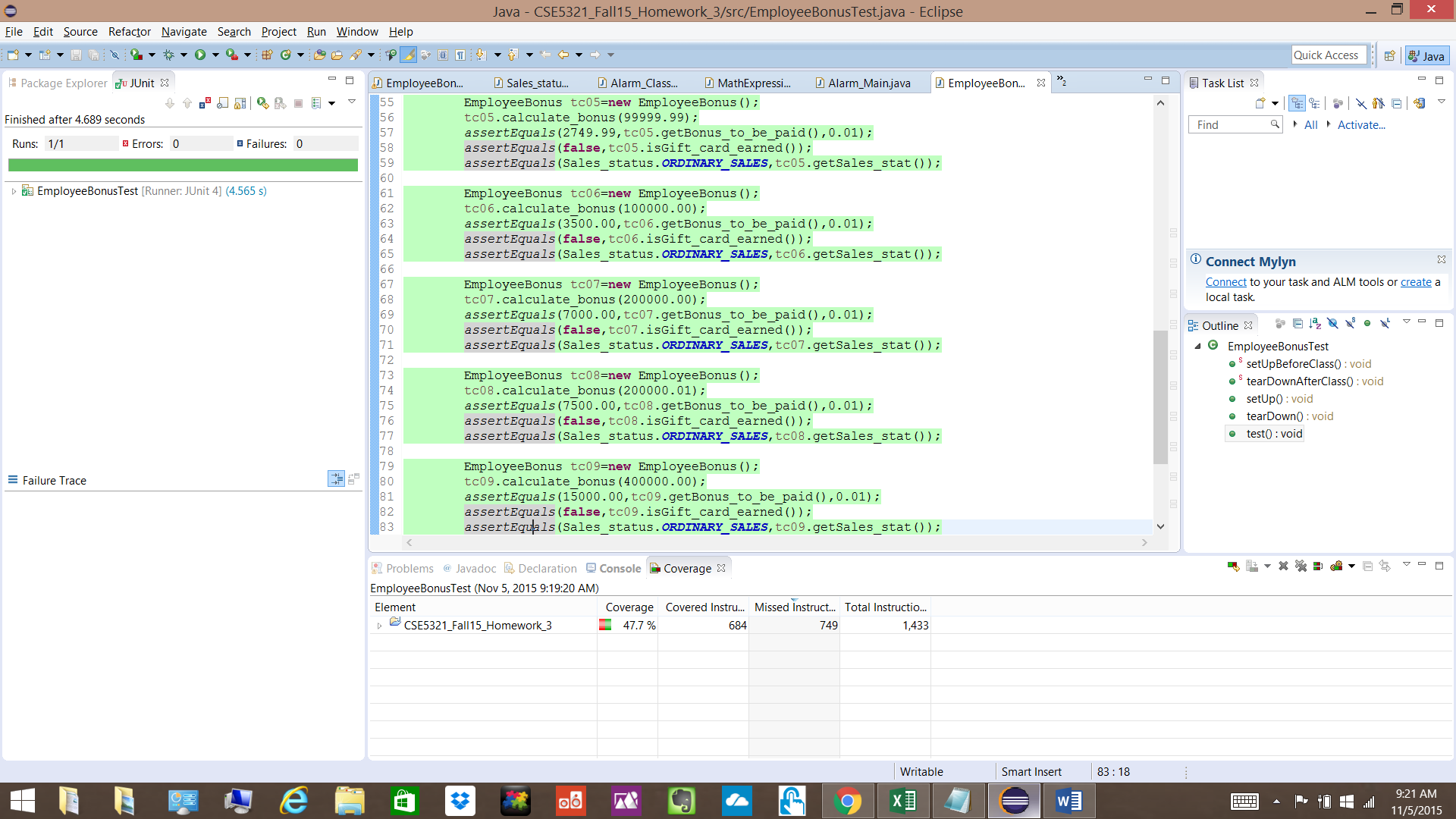
}

}

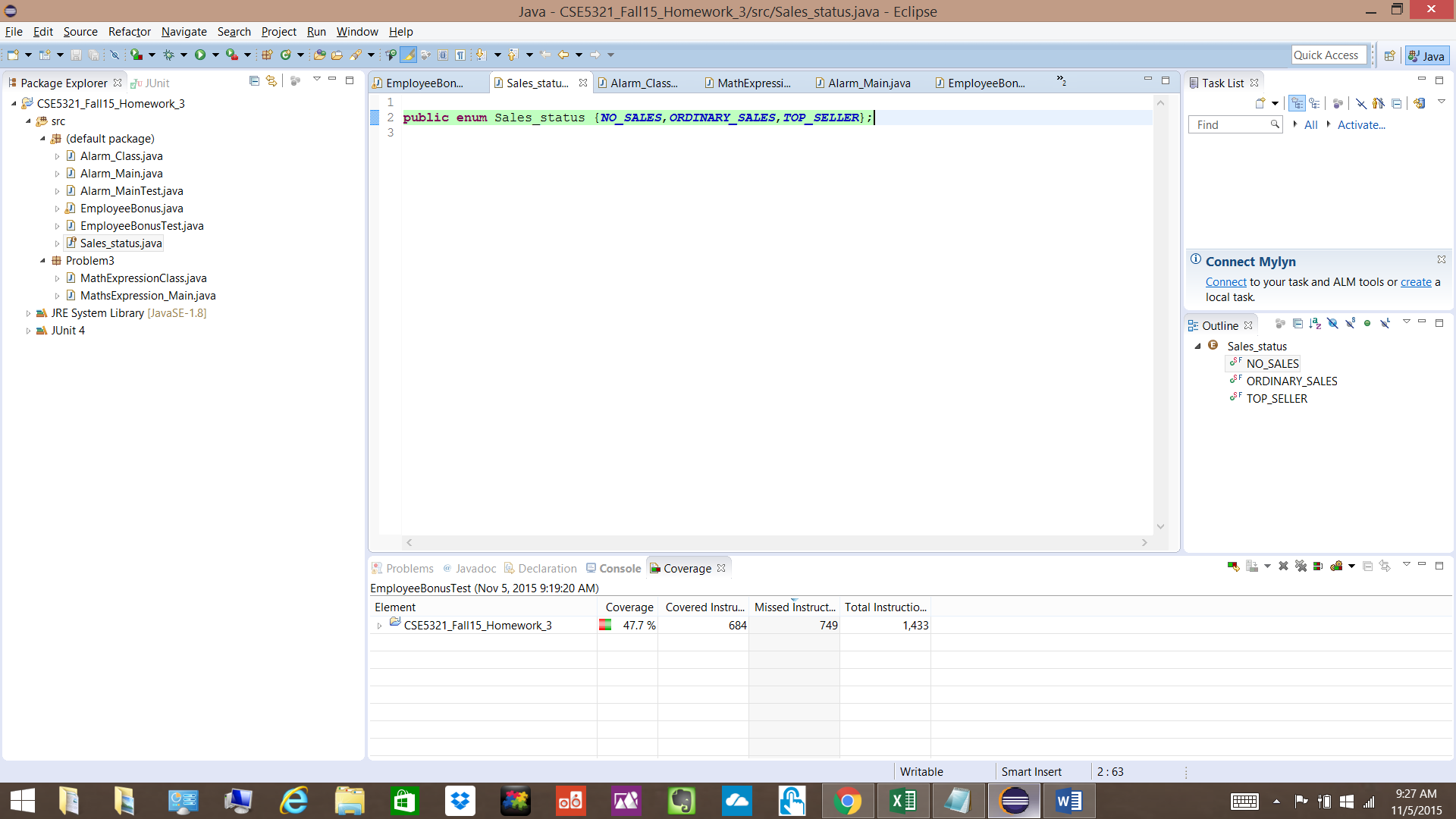
**Junit Test Cases and JACOCO coverage snapshots** given as below and src also attached.

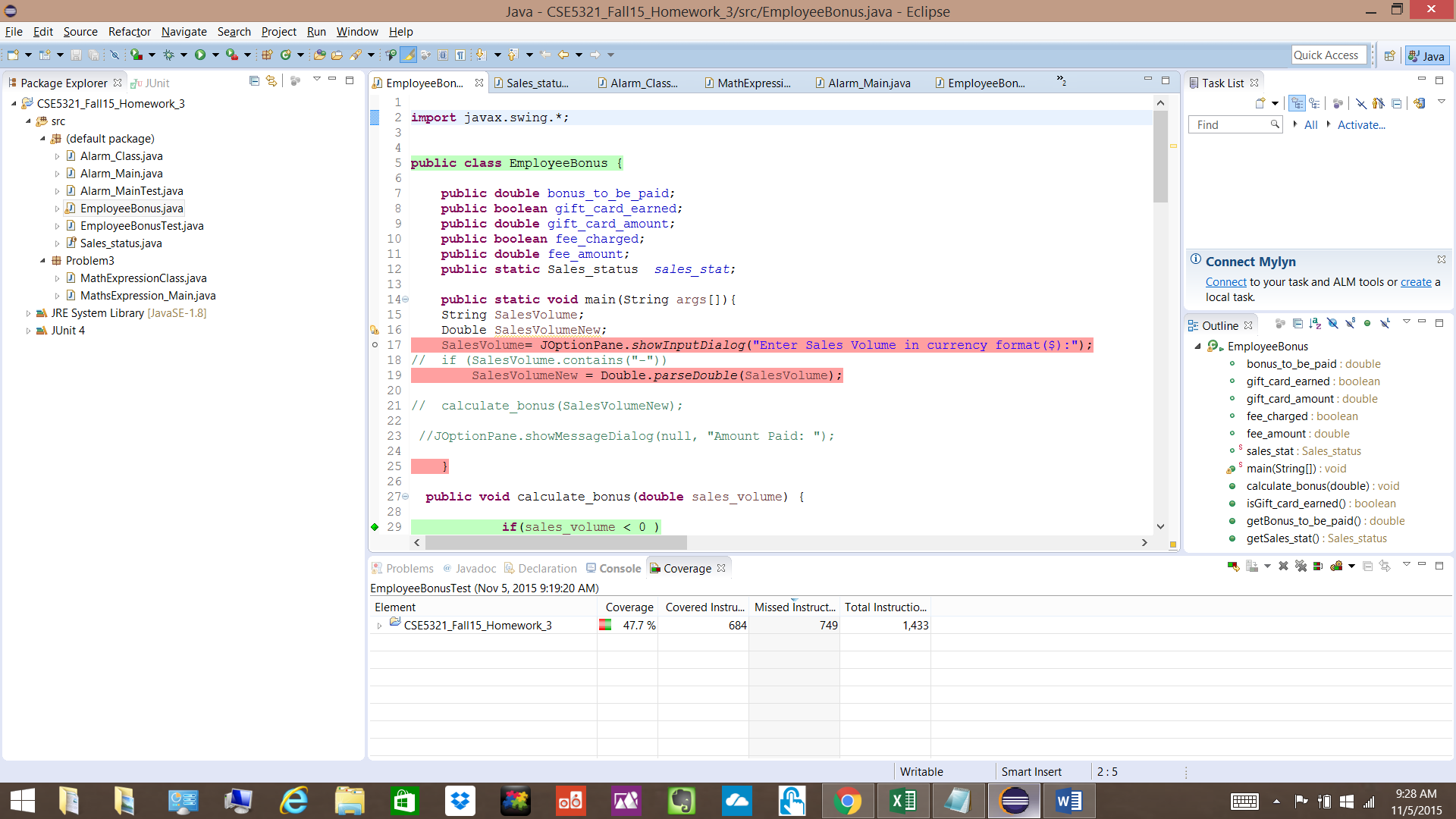


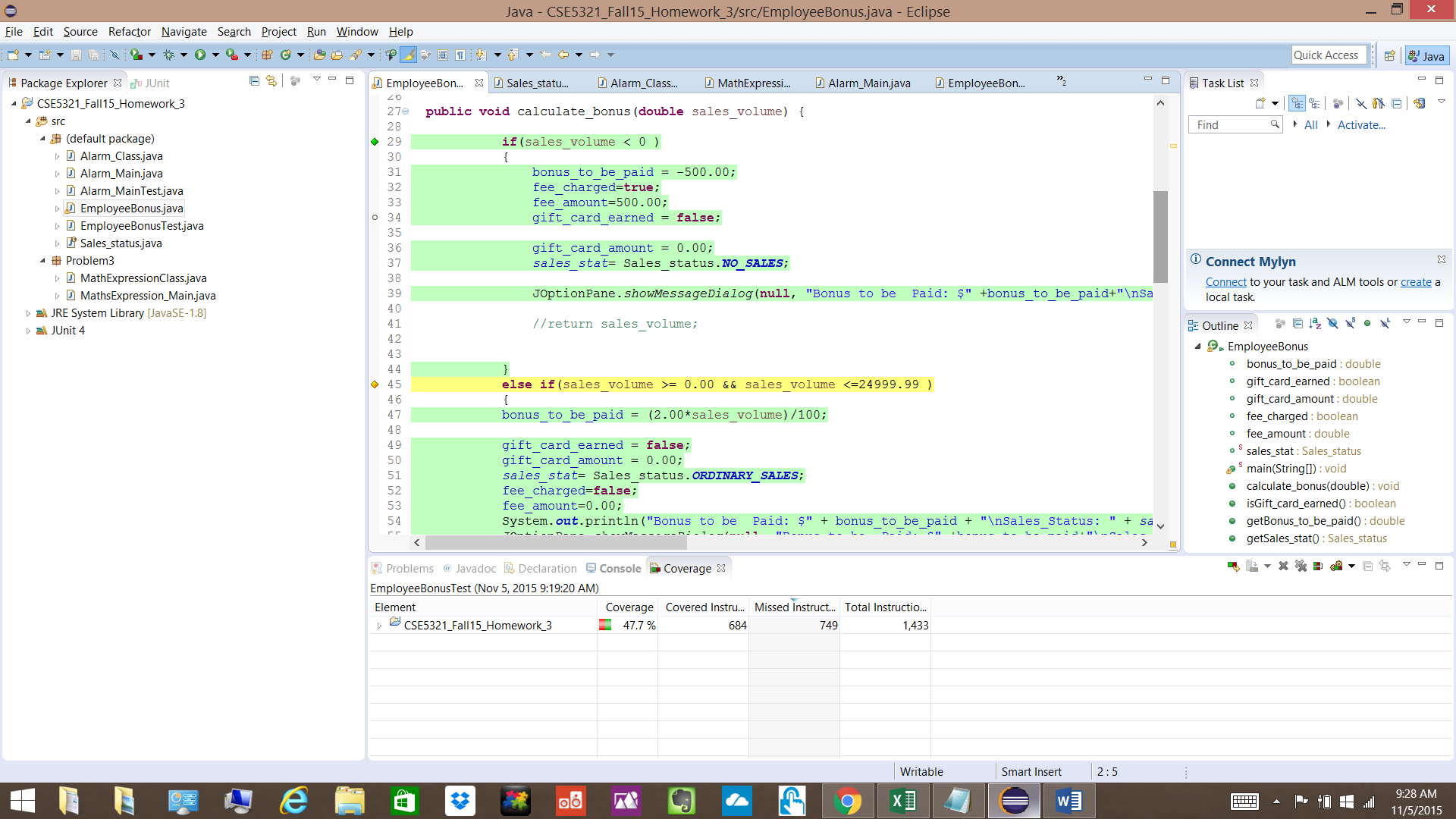


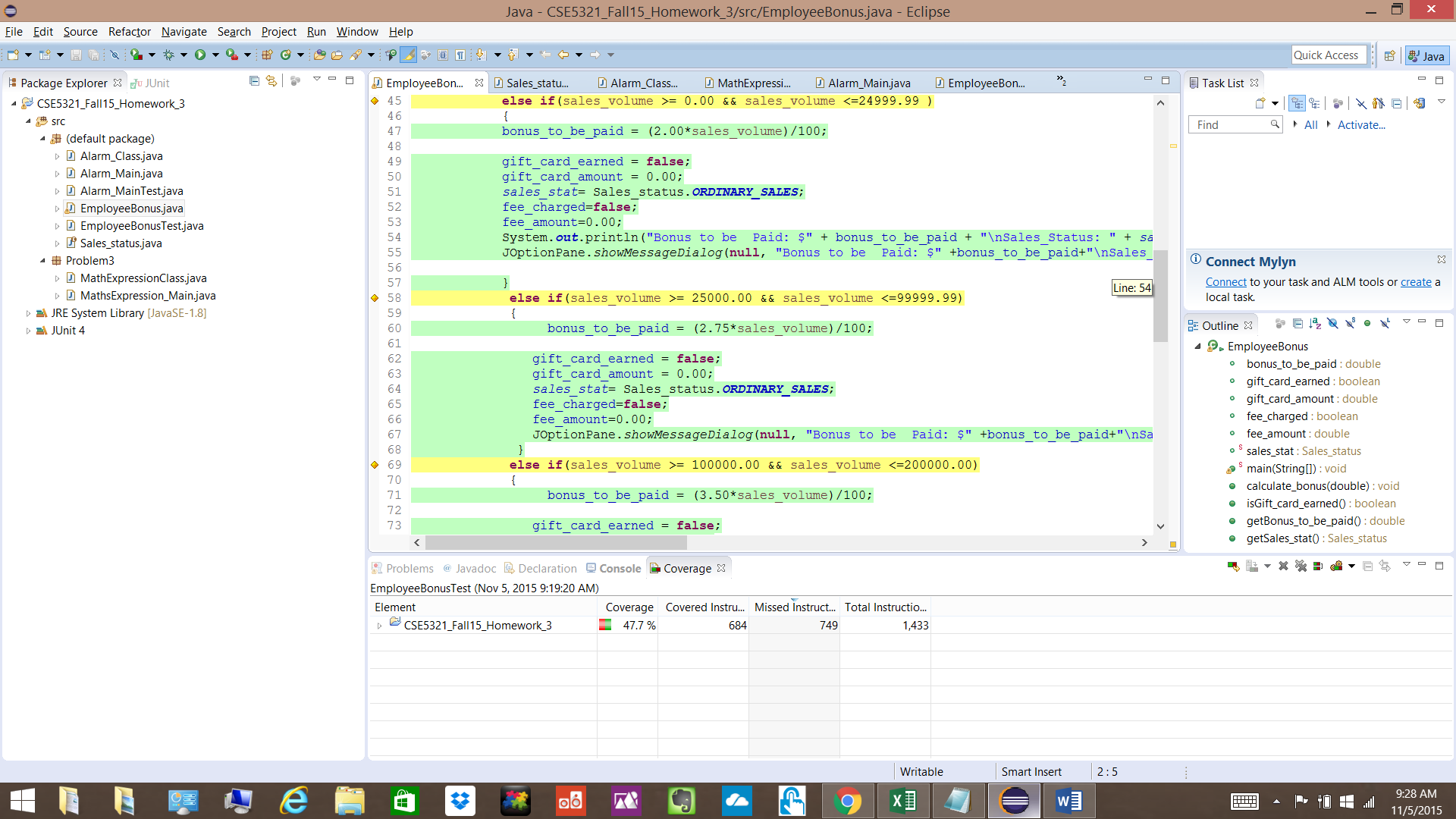


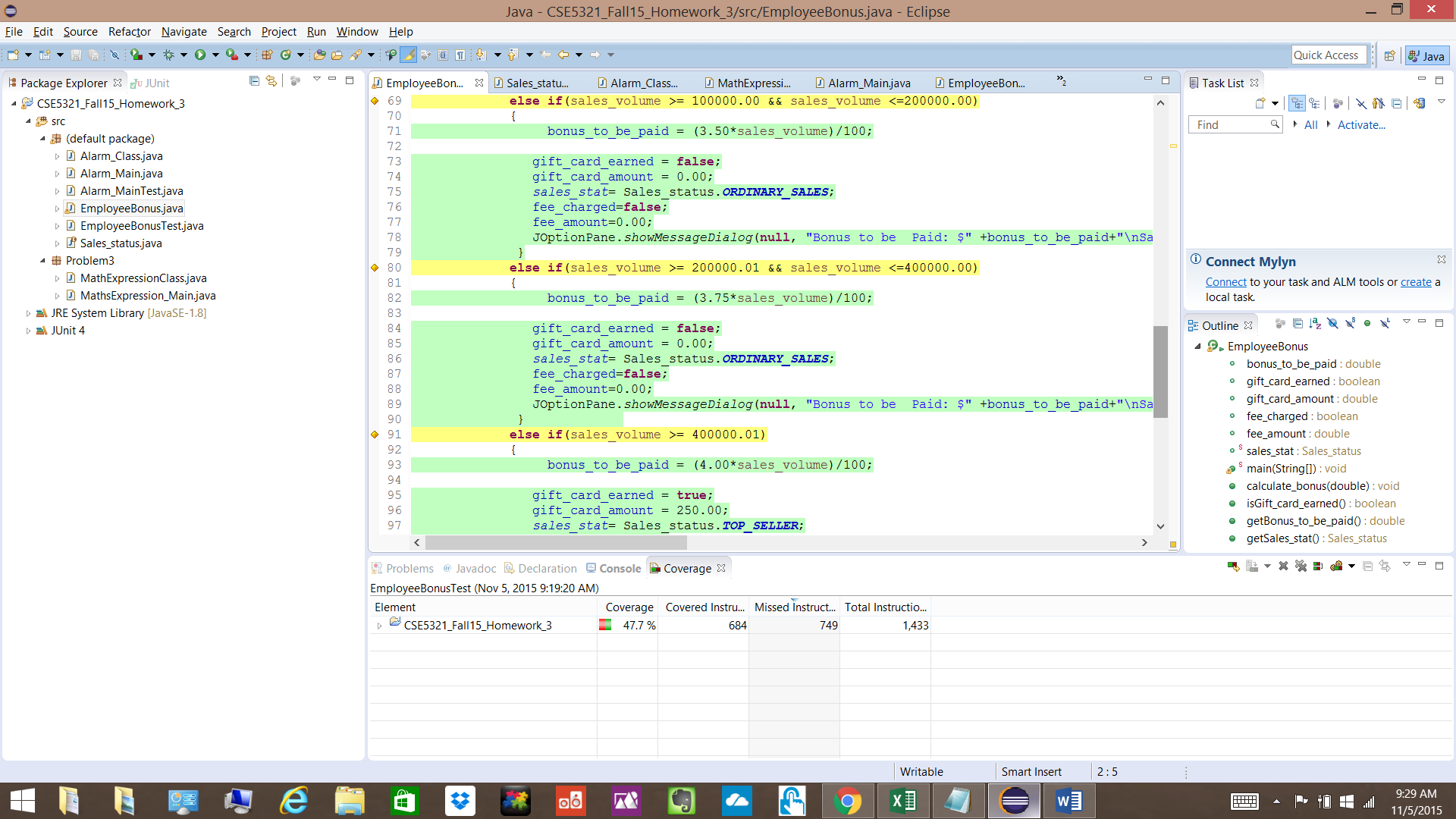


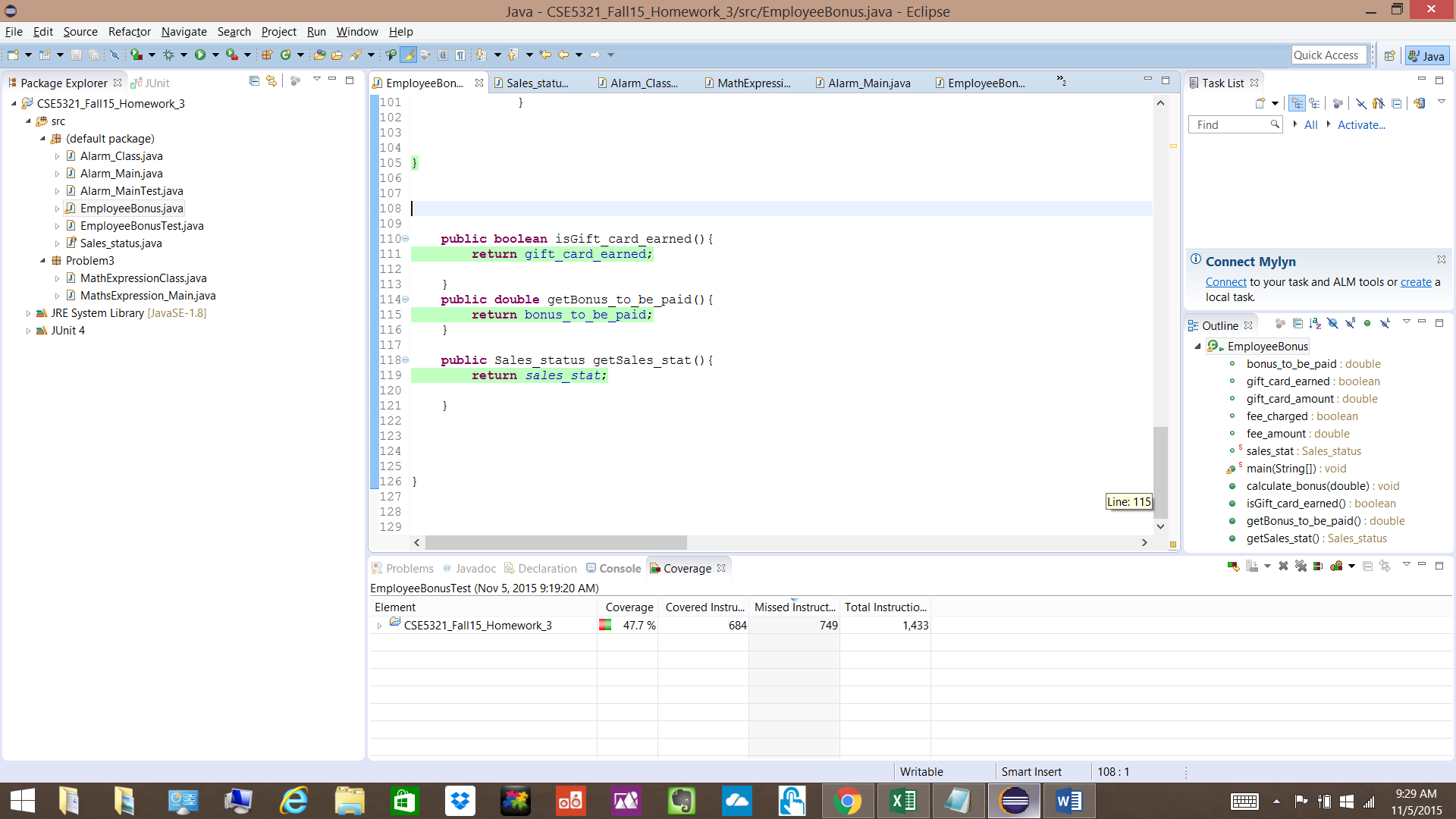
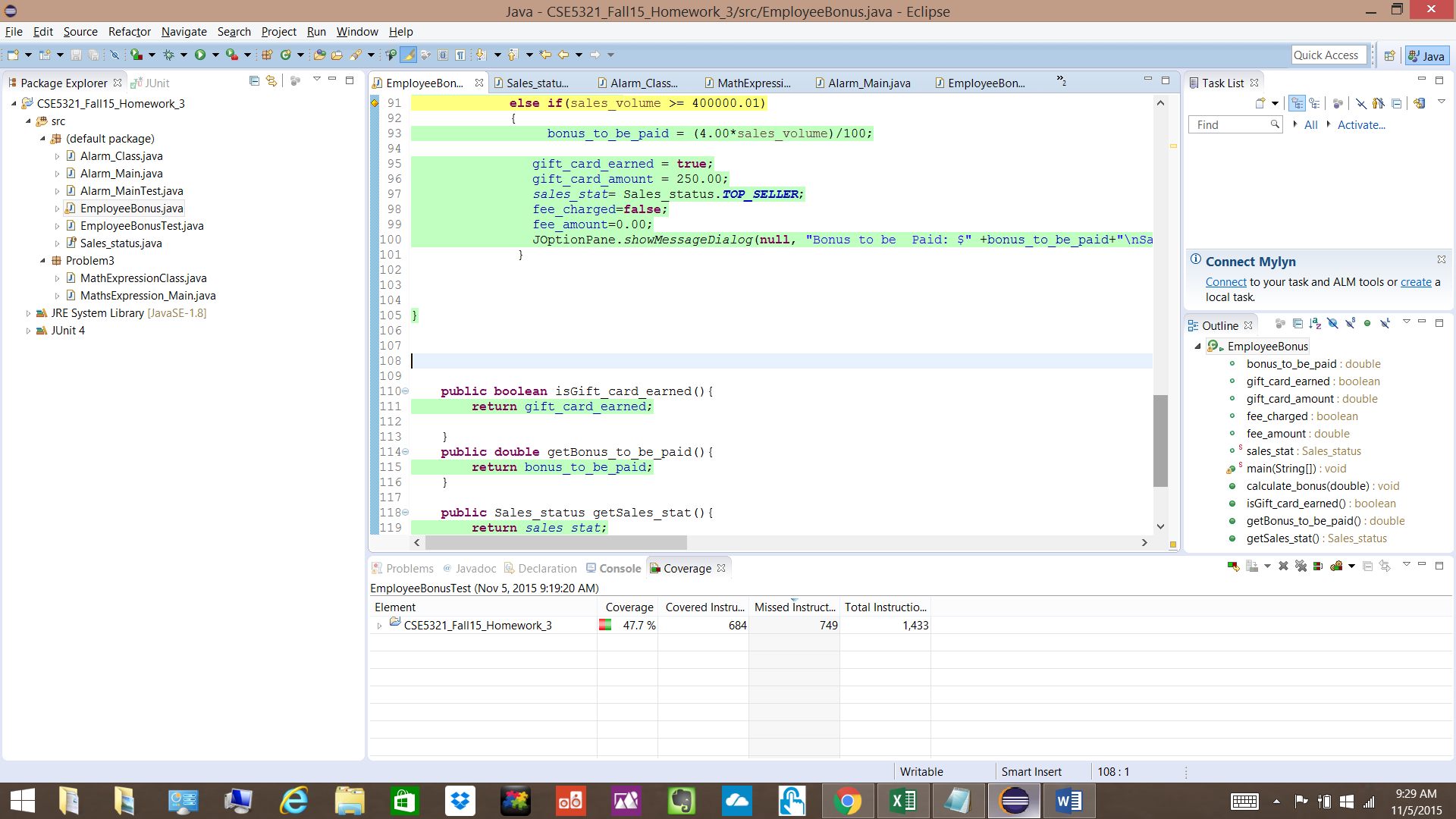












**Problem 2**

Given problem can be formulated as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tank Reading (Gallons) | Red Light | Yellow Light | Green Light | Chime |
| 500.0 | Off | Off | Off | On |
| 300.0 | Off | Off | Off | On |
| 299.9 | On | Off | Off | On |
| 200.0 | On | Off | Off | On |
| 199.9 | On | On | Off | On |
| 100.0 | On | On | Off | On |
| 99.9 | On | On | On | On |
| 50.1 | On | On | On | On |
| 50.0 | On | On | On | Off |
| 0.0 | On | On | On | Off |

**Test Case Table** for above problem is given as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Input | Expected Output | | | |
| Test Case | Tank Reading (Gallons) | Red Light | Yellow Light | Green Light | Chime |
| 1 | 500.0 | Off | Off | Off | On |
| 2 | 300.0 | Off | Off | Off | On |
| 3 | 299.9 | On | Off | Off | On |
| 4 | 200.0 | On | Off | Off | On |
| 5 | 199.9 | On | On | Off | On |
| 6 | 100.0 | On | On | Off | On |
| 7 | 99.9 | On | On | On | On |
| 8 | 50.1 | On | On | On | On |
| 9 | 50.0 | On | On | On | Off |
| 10 | 0.0 | On | On | On | Off |

**Decision Table** as below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Conditions / Rules** | 1 | 2 | 3 | 4 | 5 |
| 500.0 >= Tank Capacity >= 300.0 | Y | N | N | N | N |
| 299.9 >=Tank Capacity >= 200.0 | N | Y | N | N | N |
| 199.9 >= Tank Capacity >= 100.0 | N | N | Y | N | N |
| 99.9 >=Tank Capacity >= 50.1 | N | N | N | Y | N |
| 50.0 >=Tank Capacity >= 0.0 | N | N | N | N | Y |
| **Actions** |  |  |  |  |  |
| Red Light | Off | On | On | On | On |
| Yellow Light | Off | Off | On | On | On |
| Green Light | Off | Off | Off | On | On |
| Chime | On | On | On | On | Off |

**JAVA Code for above problem as below and also attached.**

**Alarm\_Main.java**

**import** javax.swing.JOptionPane;

**public** **class** Alarm\_Main {

**public** **static** **void** main(String args[]){

Alarm\_Class alarm = **new** Alarm\_Class();

String gas\_tank\_reading;

gas\_tank\_reading= JOptionPane.*showInputDialog*("Enter gas tank reading :");

**double** gas\_tank\_readingNew = Double.*parseDouble*(gas\_tank\_reading);

**double** gas\_tank\_capacity = 500.0;

**if**((gas\_tank\_readingNew > gas\_tank\_capacity) || (gas\_tank\_readingNew < 0.0)) {

JOptionPane.*showMessageDialog*(**null**, "Gas reading should be between 0.0 to 500.0");

} **else** {

alarm.compute\_alarms(gas\_tank\_readingNew);

}

}

}

**Alarm\_Class.java**

**import** javax.swing.\*;

**public** **class** Alarm\_Class {

**private** **boolean** red\_light\_on=**false**;

**private** **boolean** green\_light\_on=**false**;

**private** **boolean** yellow\_light\_on=**false**;

**private** **boolean** chime\_on=**false**;

**public** **void** compute\_alarms(**double** gas\_tank\_reading) {

**if** (gas\_tank\_reading >= 0.0 && gas\_tank\_reading <= 50.0)

{

red\_light\_on=**true**;

yellow\_light\_on=**true**;

green\_light\_on=**true**;

chime\_on=**false**;

JOptionPane.*showMessageDialog*(**null**,"Red\_Light On: "+red\_light\_on+"\nYellow Light On: "+yellow\_light\_on+"\nGreen Light On: "+green\_light\_on+"\nChime On:"+chime\_on);

}

**else** **if**(gas\_tank\_reading >= 50.1 && gas\_tank\_reading <= 99.9)

{

red\_light\_on=**true**;

yellow\_light\_on=**true**;

green\_light\_on=**true**;

chime\_on=**true**;

JOptionPane.*showMessageDialog*(**null**,"Red\_Light On: "+red\_light\_on+"\nYellow Light On: "+yellow\_light\_on+"\nGreen Light On: "+green\_light\_on+"\nChime On:"+chime\_on);

}

**else** **if**(gas\_tank\_reading >= 100.0 && gas\_tank\_reading <= 199.9)

{

red\_light\_on=**true**;

yellow\_light\_on=**true**;

green\_light\_on=**false**;

chime\_on=**true**;

JOptionPane.*showMessageDialog*(**null**,"Red\_Light On: "+red\_light\_on+"\nYellow Light On: "+yellow\_light\_on+"\nGreen Light On: "+green\_light\_on+"\nChime On:"+chime\_on);

}

**else** **if**(gas\_tank\_reading >= 200.0 && gas\_tank\_reading <= 299.9)

{

red\_light\_on=**true**;

yellow\_light\_on=**false**;

green\_light\_on=**false**;

chime\_on=**true**;

JOptionPane.*showMessageDialog*(**null**,"Red\_Light On: "+red\_light\_on+"\nYellow Light On: "+yellow\_light\_on+"\nGreen Light On: "+green\_light\_on+"\nChime On:"+chime\_on);

}

**else** **if**(gas\_tank\_reading >= 300.0 && gas\_tank\_reading <=500.0)

{

red\_light\_on=**false**;

yellow\_light\_on=**false**;

green\_light\_on=**false**;

chime\_on=**true**;

JOptionPane.*showMessageDialog*(**null**,"Red\_Light On: "+red\_light\_on+"\nYellow Light On: "+yellow\_light\_on+"\nGreen Light On: "+green\_light\_on+"\nChime On:"+chime\_on);

}

}

**public** **void** setRed\_light\_on(**boolean** red\_light\_on){

**this**.red\_light\_on=red\_light\_on;

}

**public** **boolean** isRed\_light\_on(){

**return** red\_light\_on;

}

**public** **void** setGreen\_light\_on(**boolean** green\_light\_on){

**this**.green\_light\_on=green\_light\_on;

}

**public** **boolean** isGreen\_light\_on(){

**return** green\_light\_on;

}

**public** **void** setYellow\_light\_on(**boolean** yellow\_light\_on){

**this**.yellow\_light\_on=yellow\_light\_on;

}

**public** **boolean** isYellow\_light\_on(){

**return** yellow\_light\_on;

}

**public** **void** setChime\_on(**boolean** chime\_on){

**this**.chime\_on=chime\_on;

}

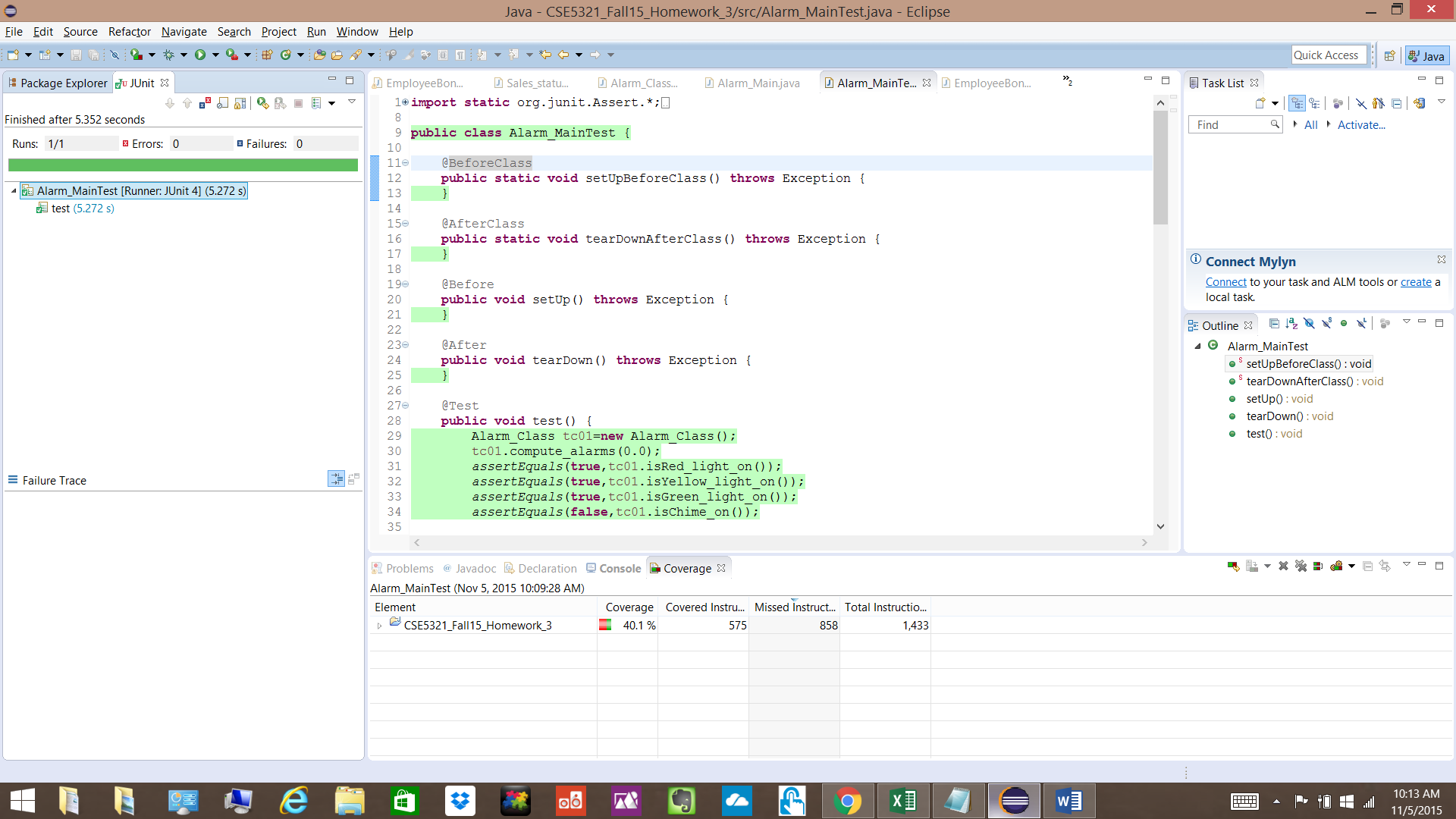
**public** **boolean** isChime\_on(){

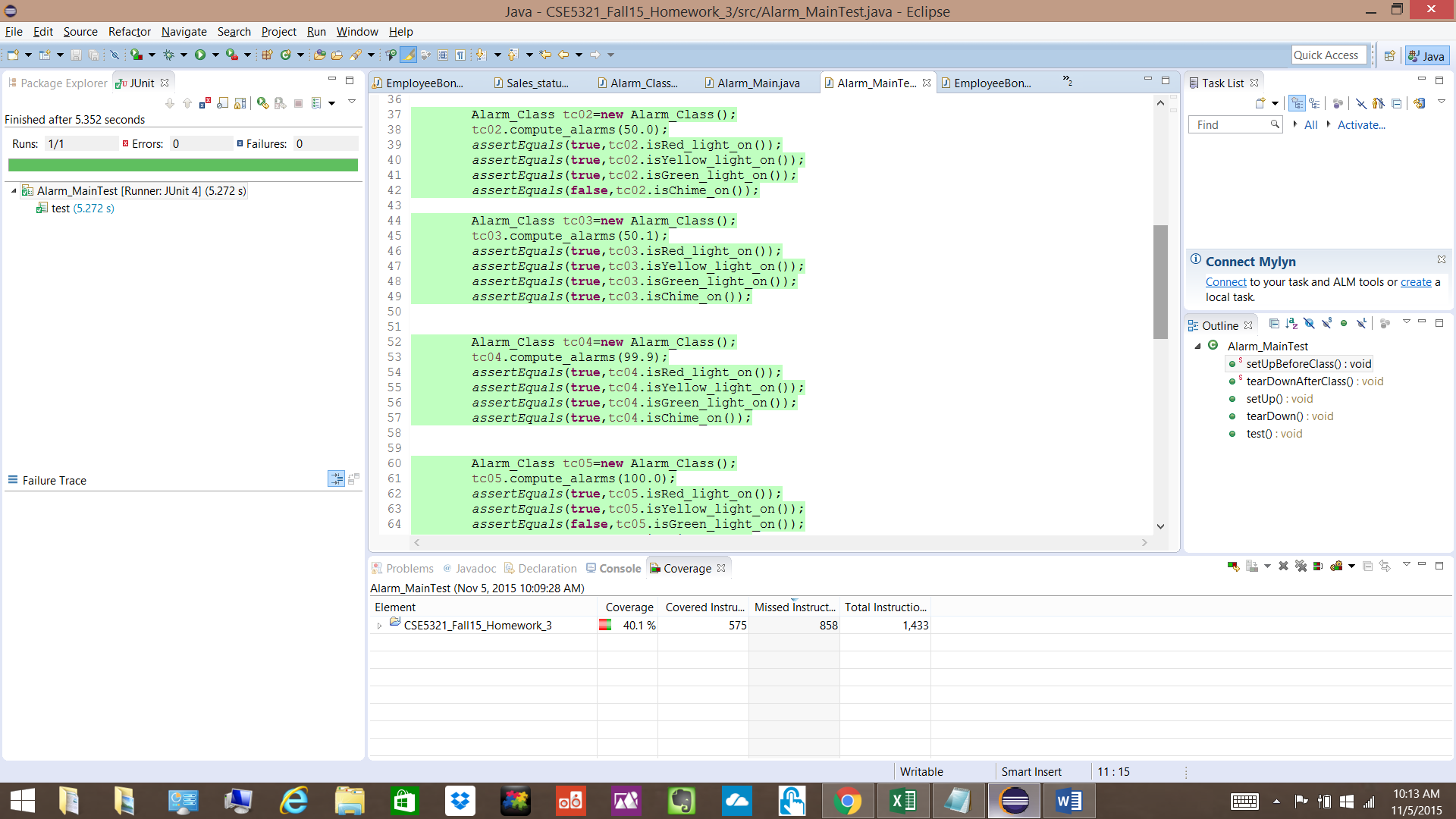
**return** chime\_on;

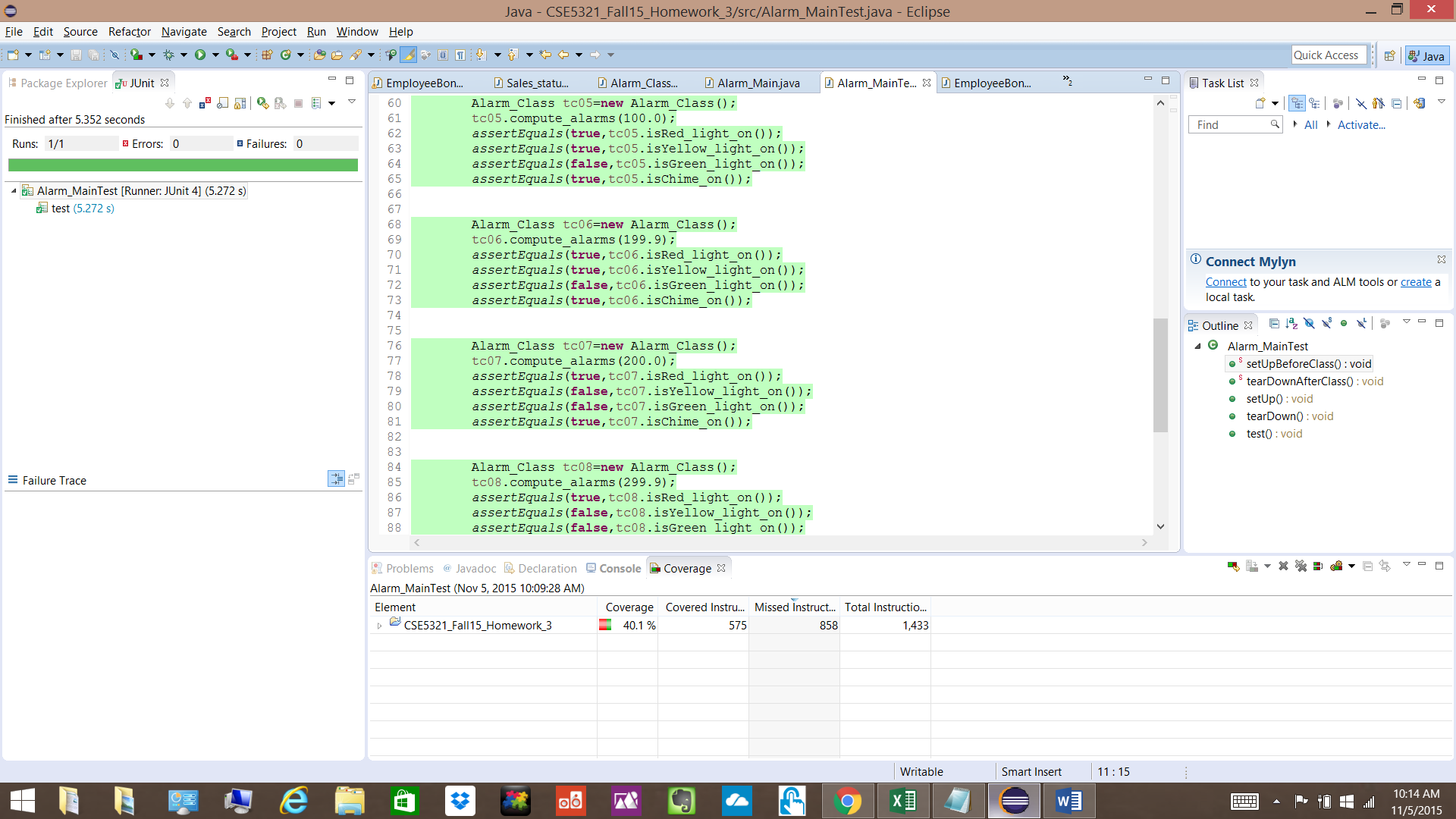
}

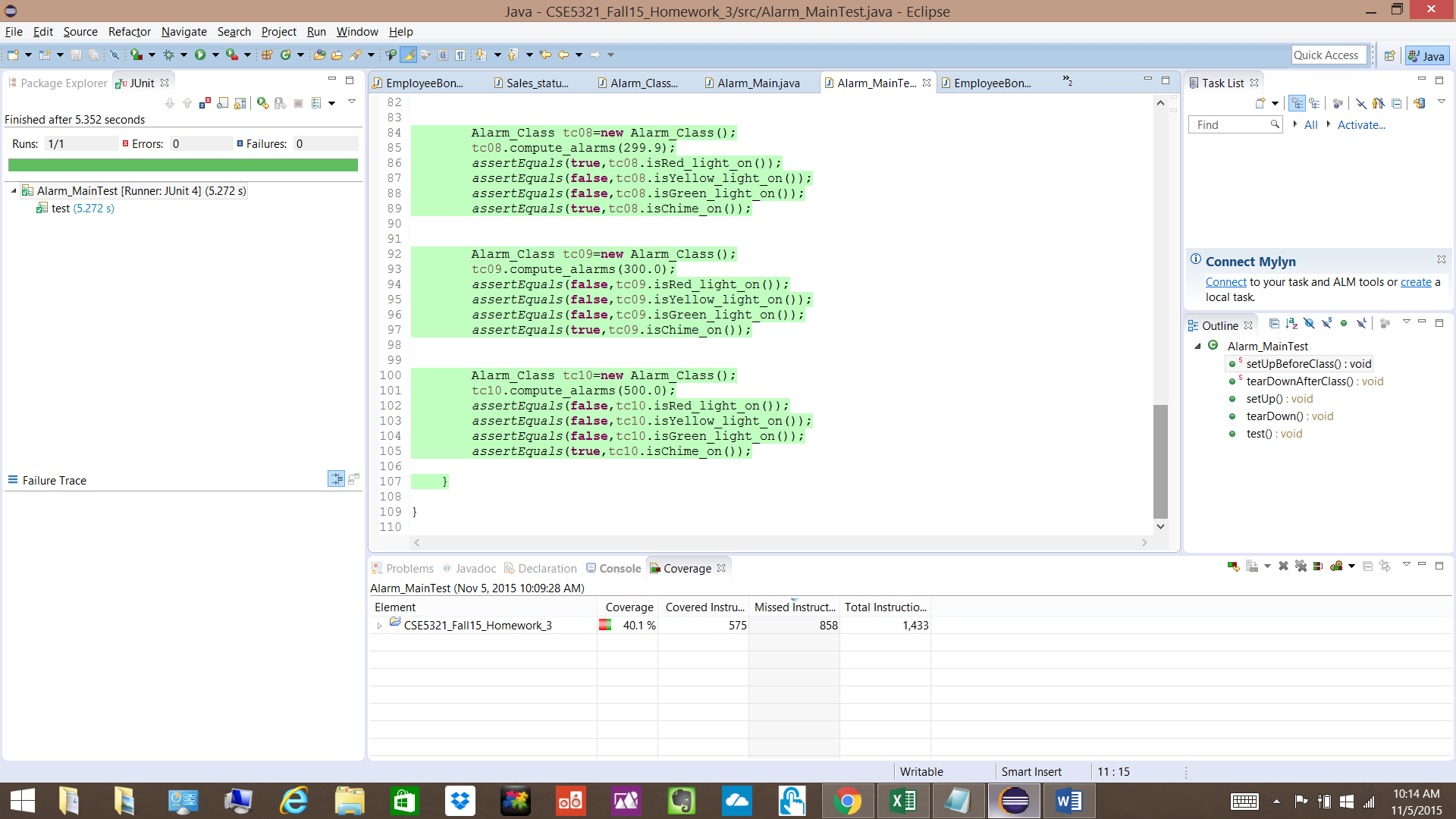
}

**Junit code and JACOCO coverage analysis snapshots as given below:**

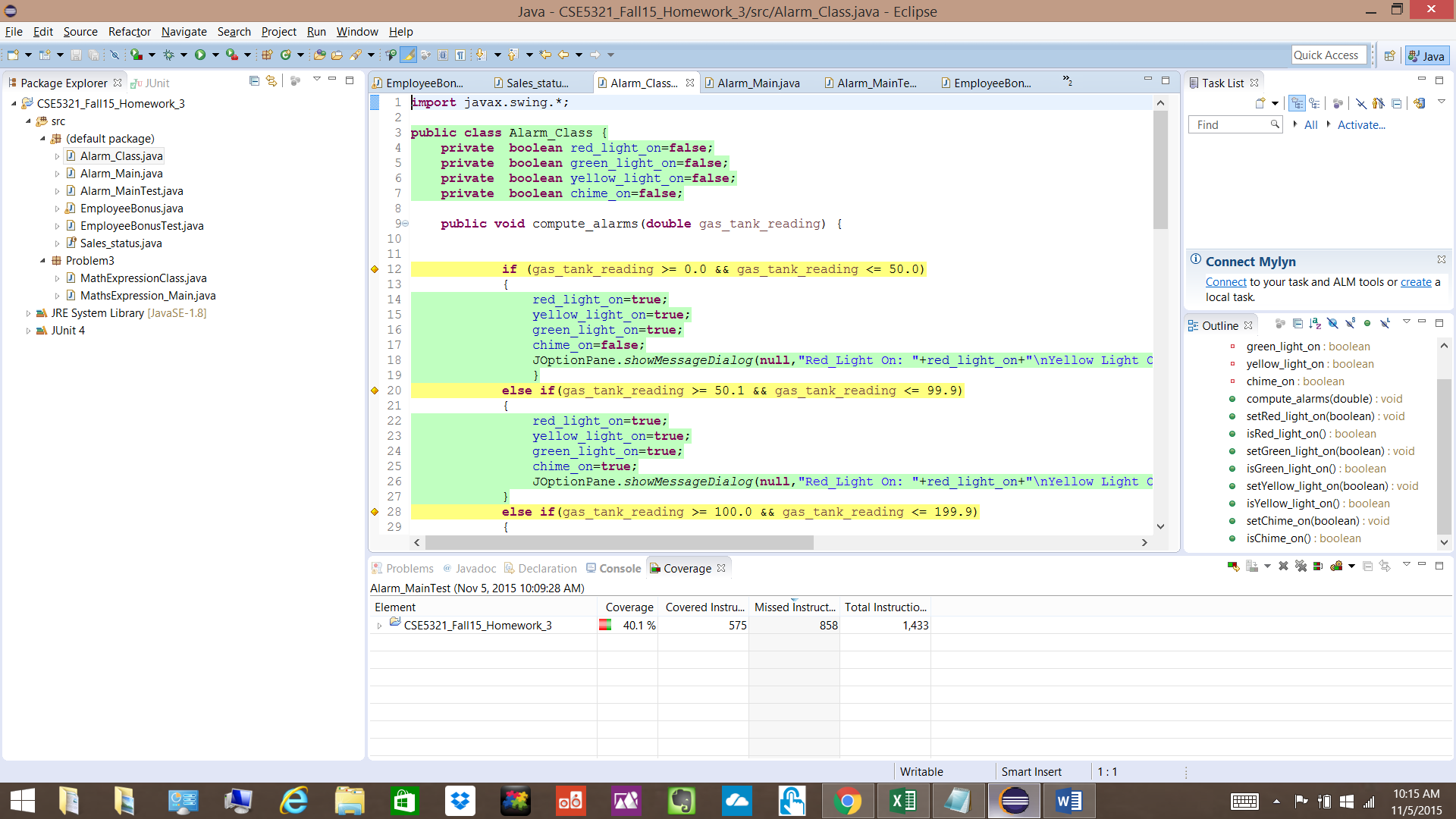


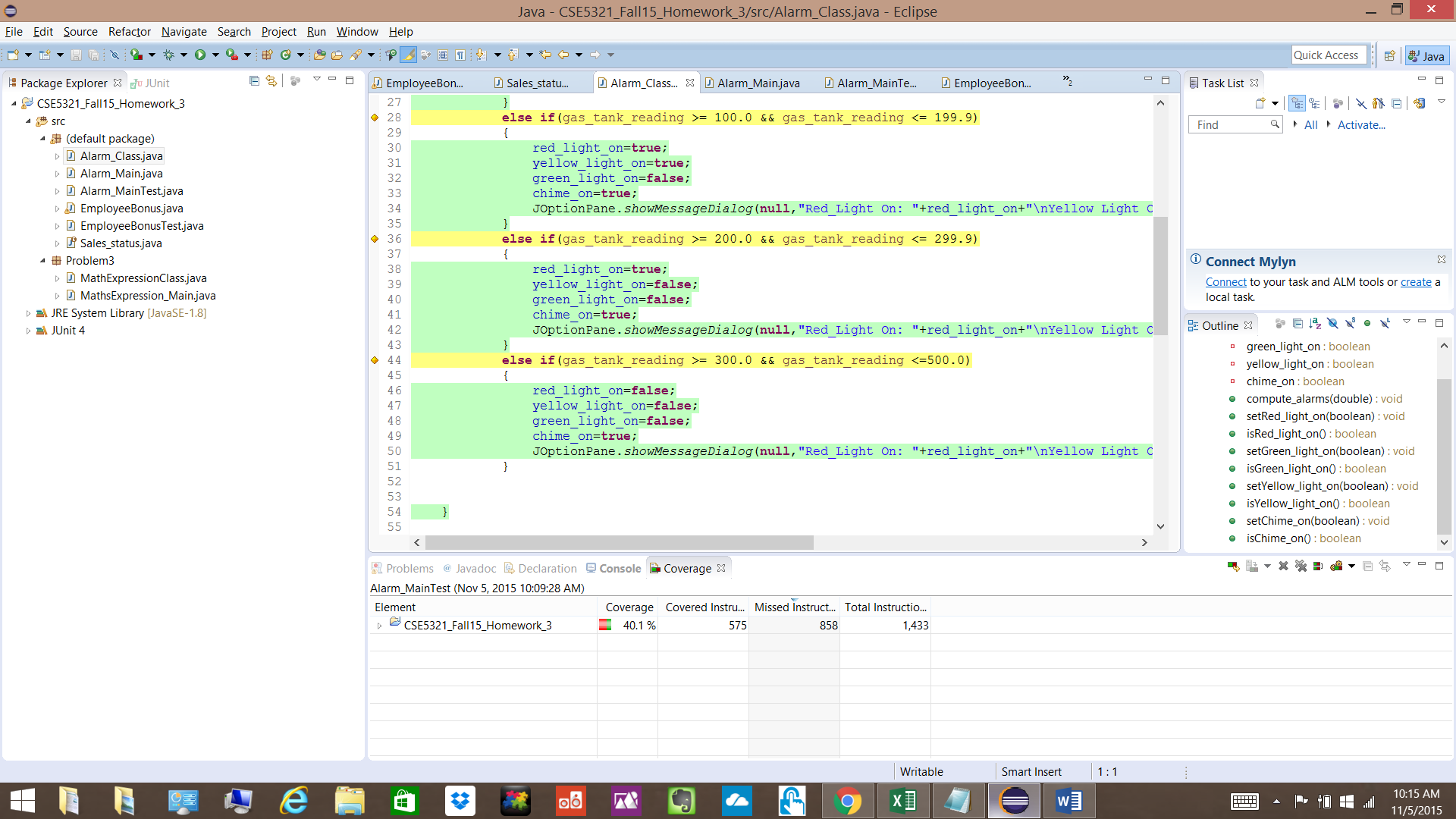


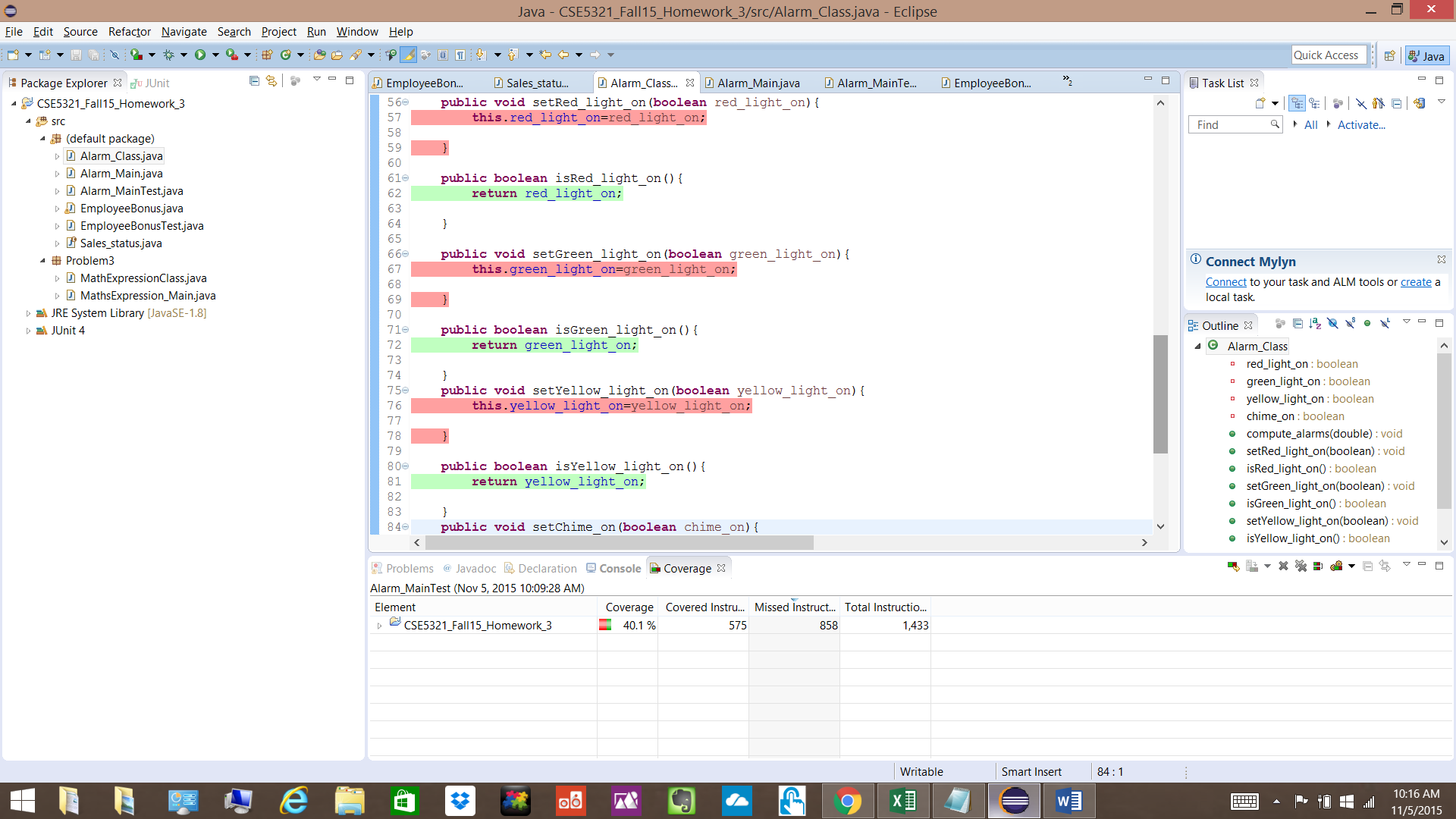


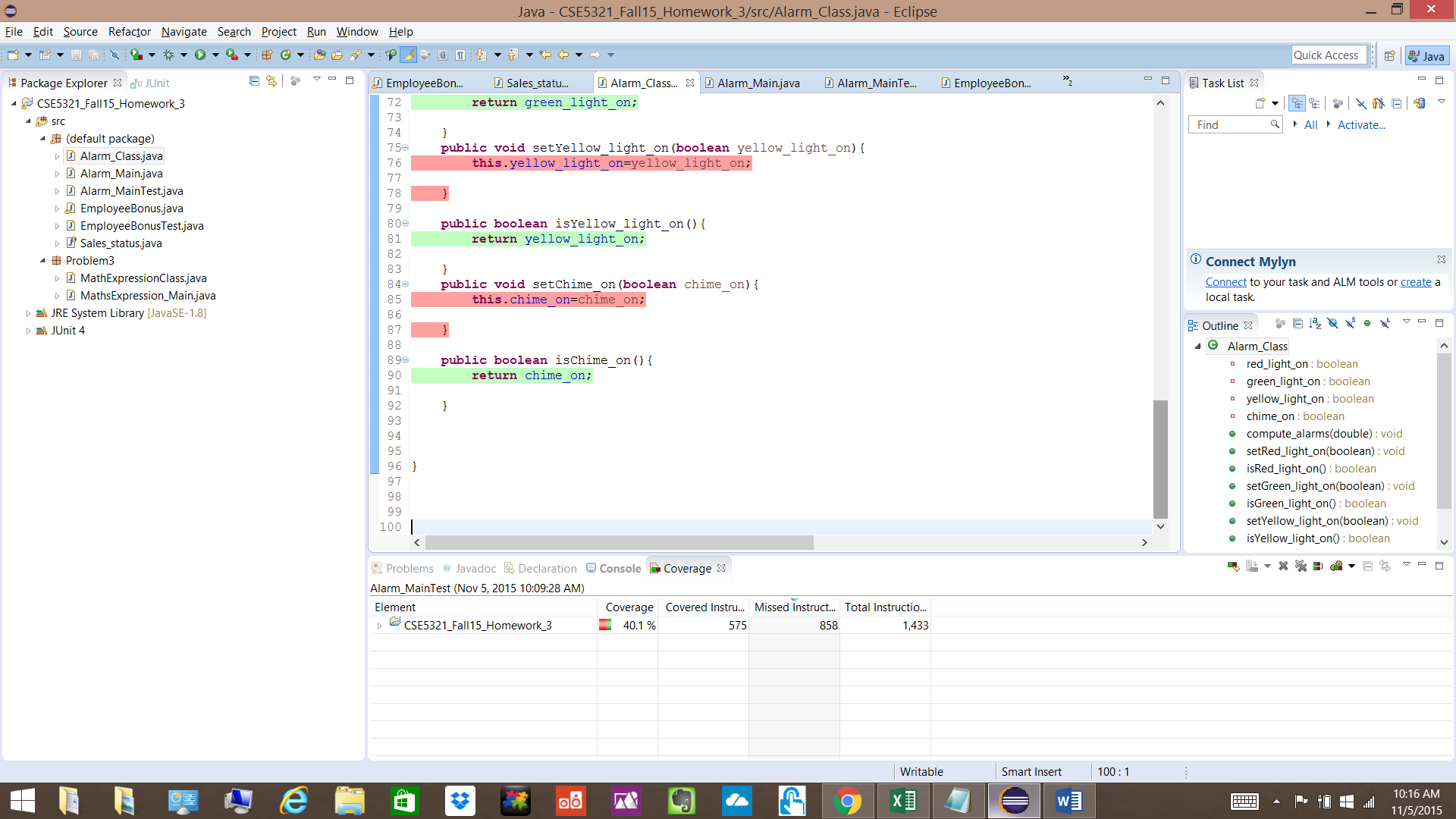








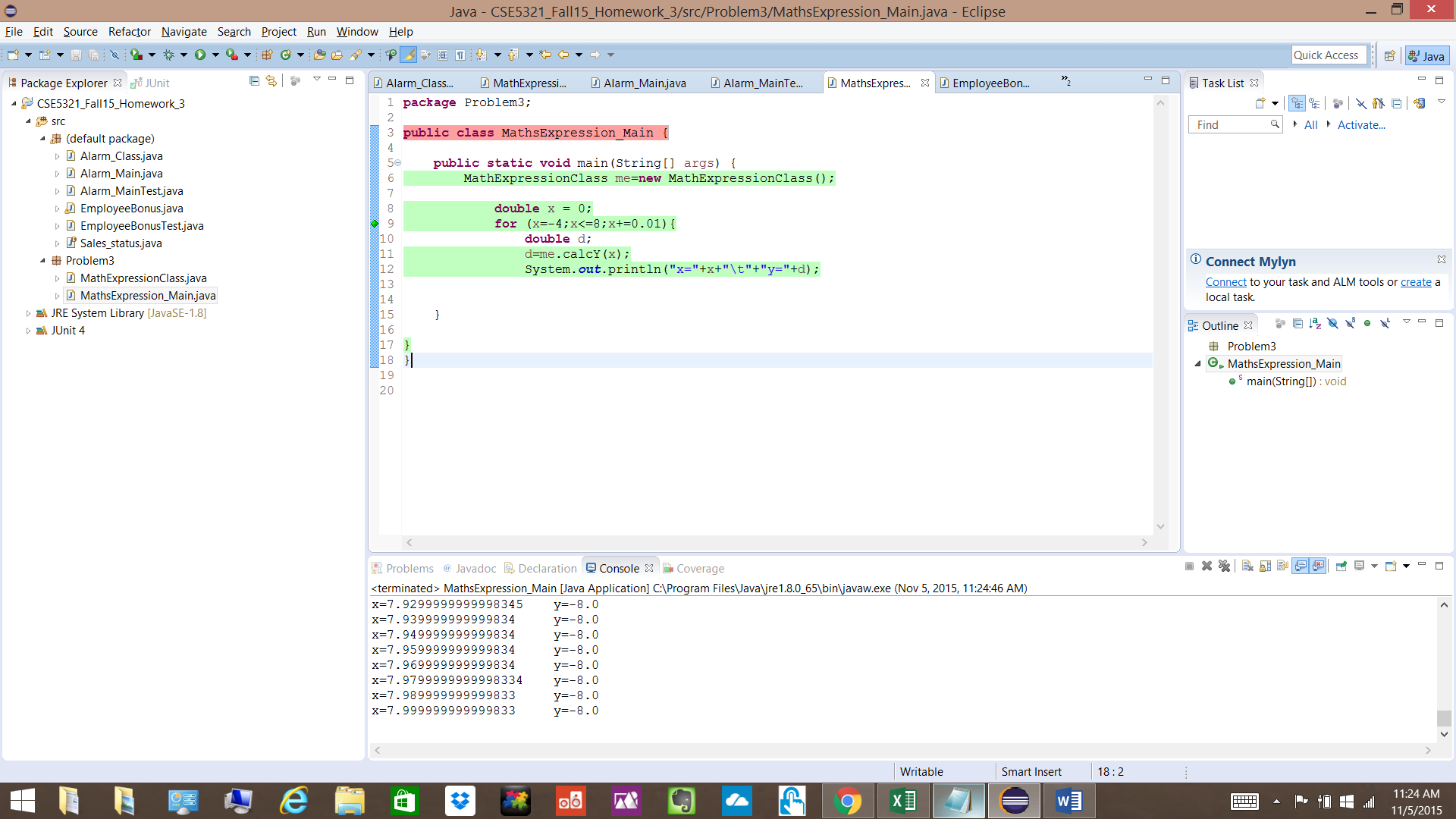




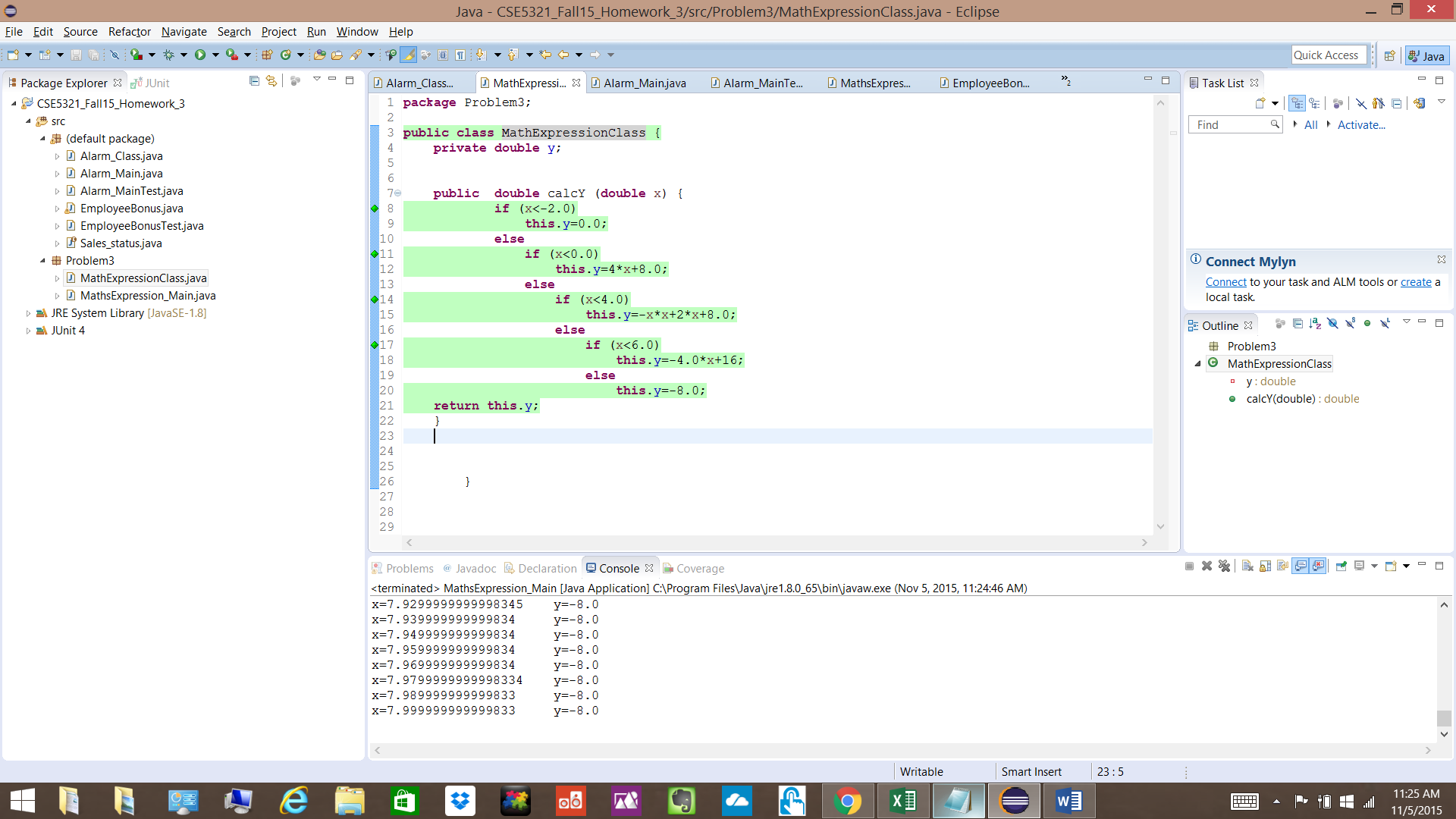
**Problem 3**

MathExpression Class Code

**MathsExpression\_Main.java**



**MathExpressionClass.java**



**Output of this code ,the plot, CFG and Test Case Table is attached in the Excel Sheet “HW3\_Problem3.xlsx”**