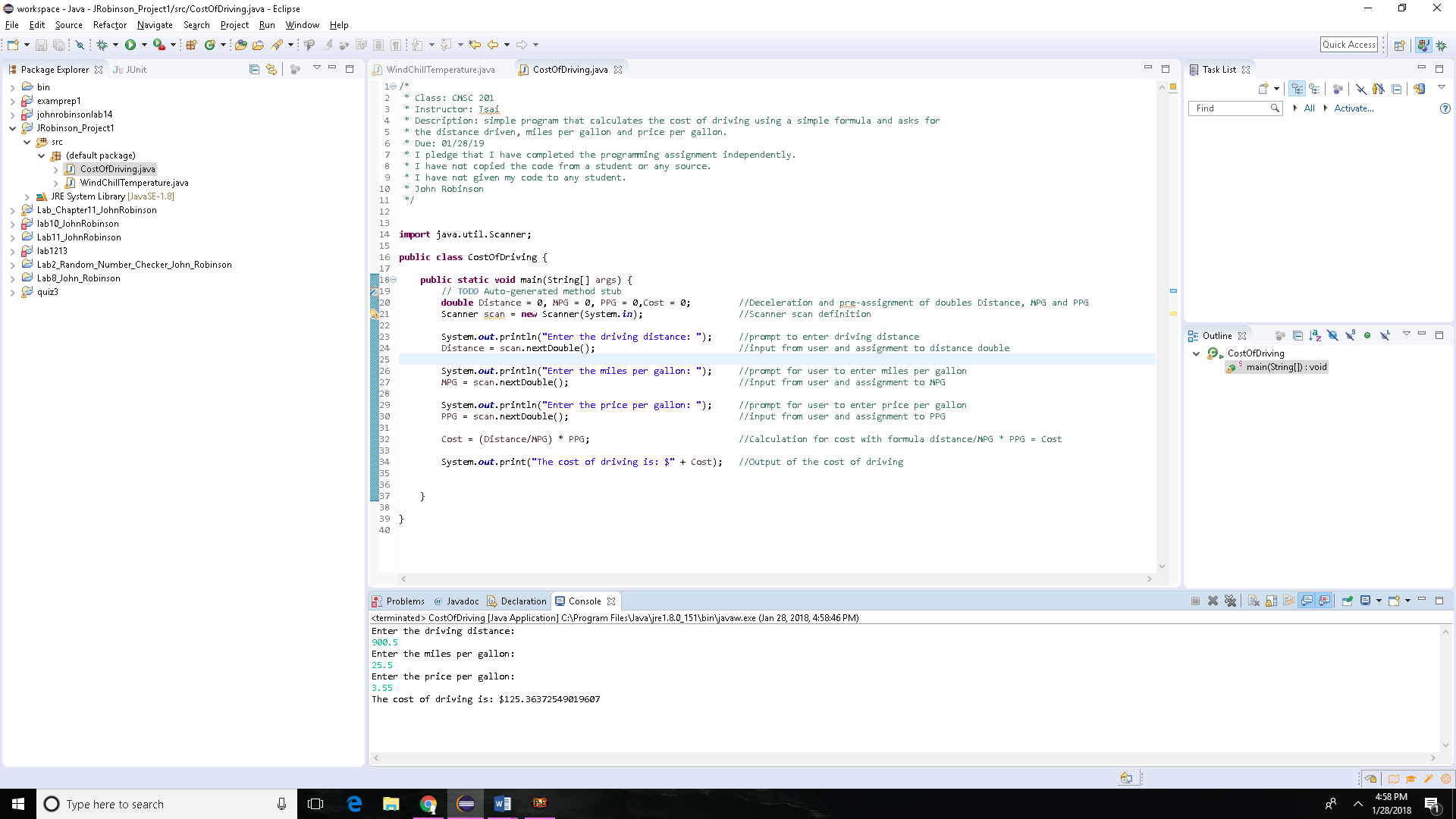
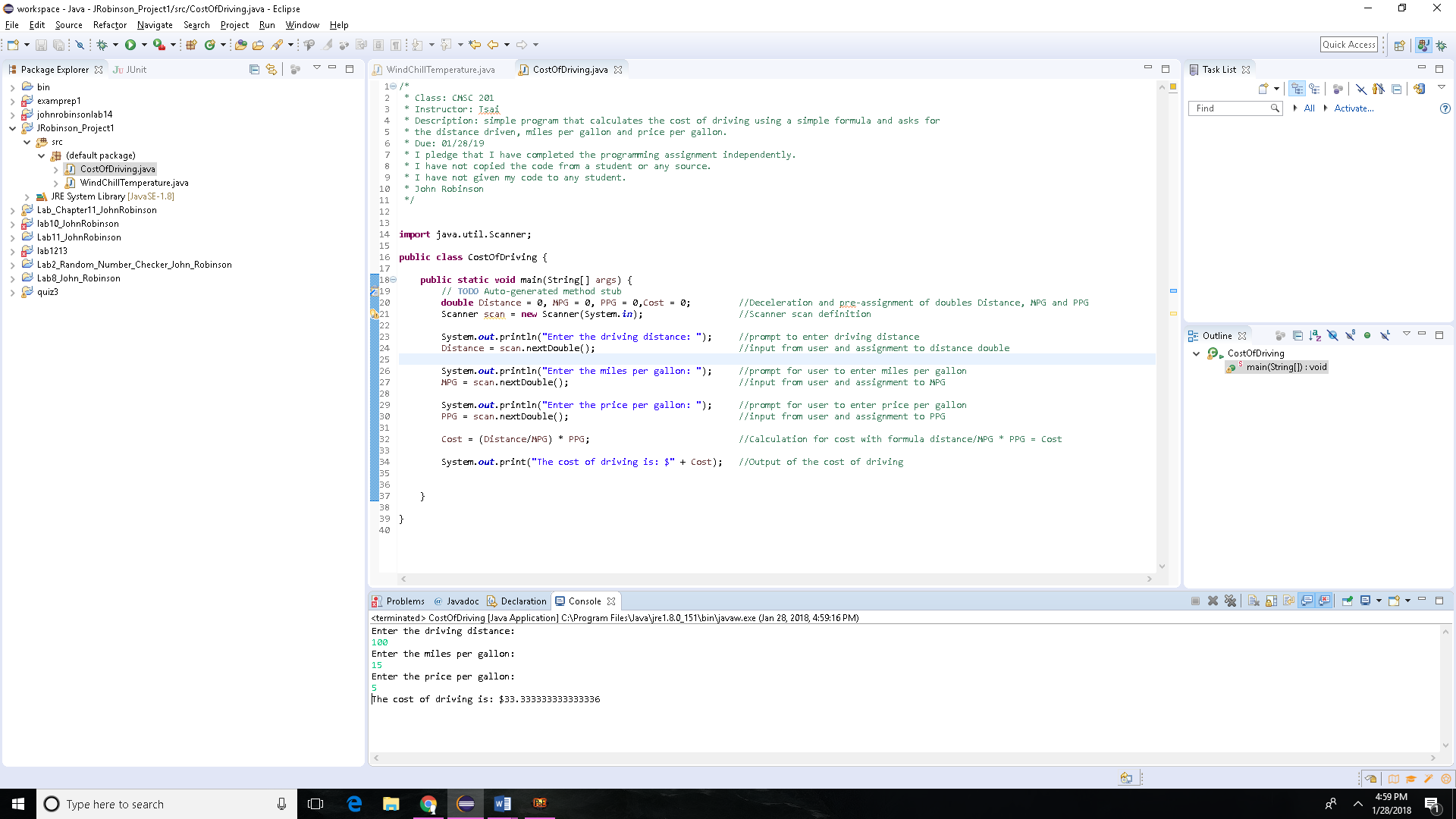
John Robinson Cost of Driving Part 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cases | Input | Expected Output | Actual Output | Pass/Fail? |
| Case 1 | Enter the driving distance:  900.5  Enter the miles per gallon:  25.5  Enter the price per gallon:  3.55 | The cost of driving is: $125.36372 | The cost of driving is: $125.36372 | Y |
| Case 2 | Enter the driving distance:  100  Enter the miles per gallon:  15  Enter the price per gallon:  5 | The cost of driving is: $33.333333 | The cost of driving is: $33.333333 | Y |
| Case 3 | Enter the driving distance:  -5  Enter the miles per gallon:  50  Enter the price per gallon:  1 | The cost of driving is: $-0.1 | The cost of driving is: $-0.1 | Y, but didn’t have test for negative distance |
| Case 4 | Enter the driving distance:  1000  Enter the miles per gallon:  -10  Enter the price per gallon:  5 | The cost of driving is: $-500.0 | The cost of driving is: $-500.0 | Y, but didn’t have test for negative miles per gallon |

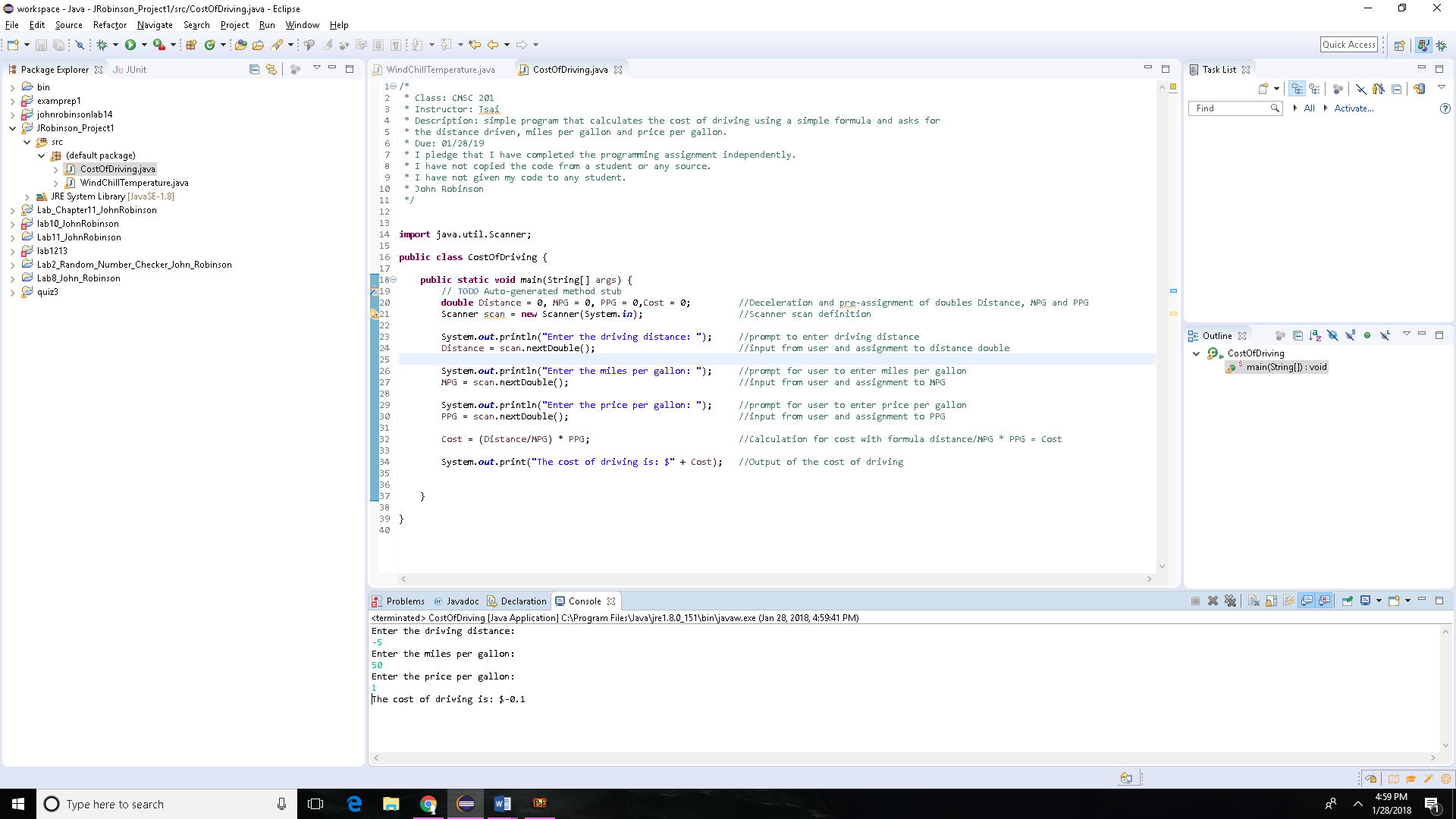
Case 1

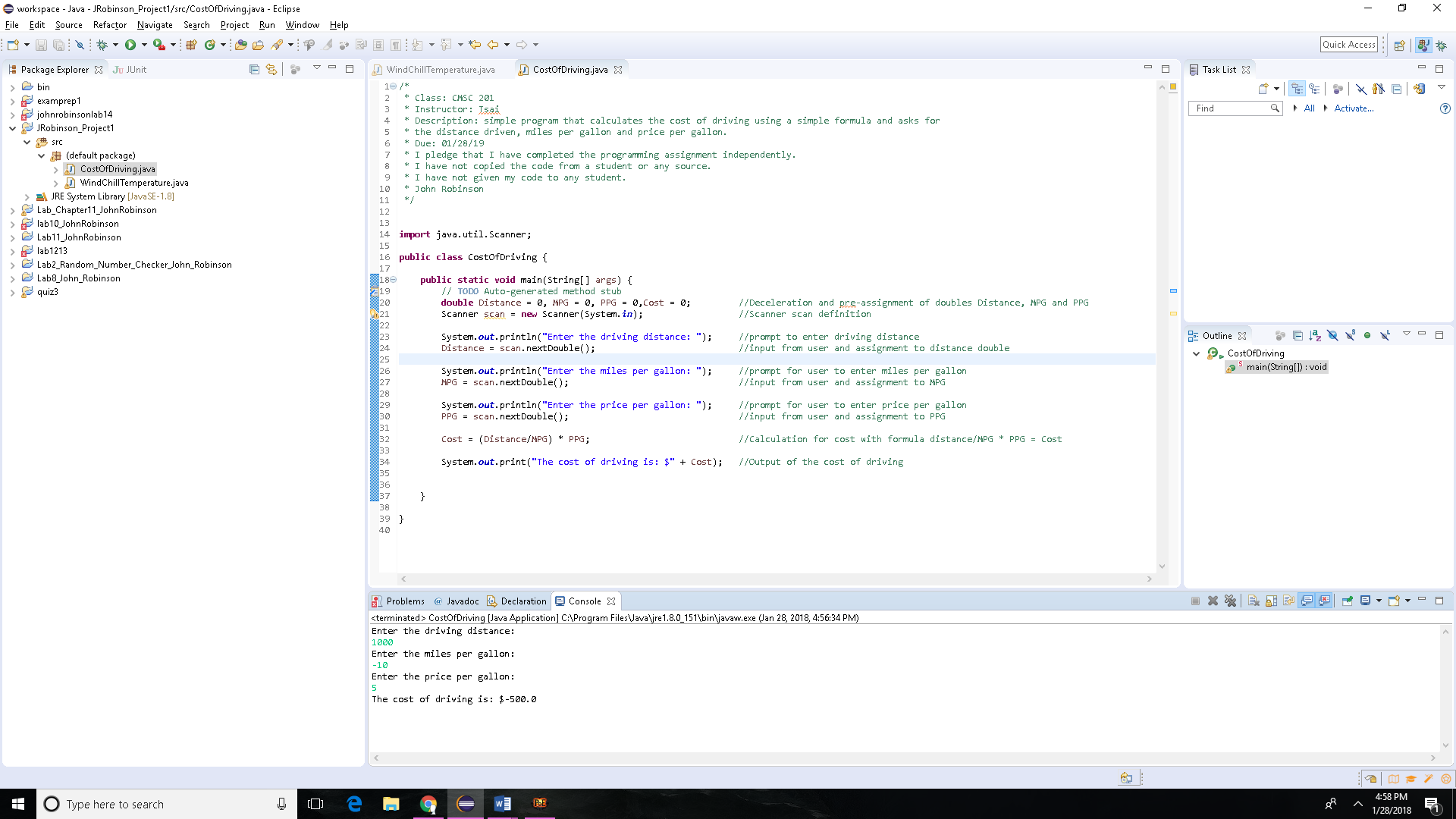


Case 2



Case 3



Case 4

UML Diagram

|  |
| --- |
| Cost of Driving |
| +Distance: double  +MPG: double  +PPG: double  +Cost: Double |
| +CostOfDriving: void |

Pseudo Code:

1. Create a scanner to receive input from the keyboard from the user
2. Declare 4 variables to store user input and give output
   1. Distance
   2. MPG
   3. PPG
   4. Cost
3. Prompt user to enter values for each apart from cost accordingly
4. Receive the data and store accordingly
5. Using this formula calculate the cost Cost = (Distance/MPG) \* PPG
6. Output the cost

Source Code:

/\*

\* Class: CMSC 201

\* Instructor: Tsai

\* Description: simple program that calculates the cost of driving using a simple formula and asks for

\* the distance driven, miles per gallon and price per gallon.

\* Due: 01/28/19

\* I pledge that I have completed the programming assignment independently.

\* I have not copied the code from a student or any source.

\* I have not given my code to any student.

\* John Robinson

\*/

**import** java.util.Scanner;

**public** **class** CostOfDriving {

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

// **TODO** Auto-generated method stub

**double** Distance = 0, MPG = 0, PPG = 0,Cost = 0; //Deceleration and pre-assignment of doubles Distance, MPG and PPG

Scanner scan = **new** Scanner(System.***in***); //Scanner scan definition

System.***out***.println("Enter the driving distance: "); //prompt to enter driving distance

Distance = scan.nextDouble(); //input from user and assignment to distance double

System.***out***.println("Enter the miles per gallon: "); //prompt for user to enter miles per gallon

MPG = scan.nextDouble(); //input from user and assignment to MPG

System.***out***.println("Enter the price per gallon: "); //prompt for user to enter price per gallon

PPG = scan.nextDouble(); //input from user and assignment to PPG

Cost = (Distance/MPG) \* PPG; //Calculation for cost with formula distance/MPG \* PPG = Cost

System.***out***.print("The cost of driving is: $" + Cost); //Output of the cost of driving

}

}

Lessons learned:

* To effectively use the scanner function in java and how to do simple calculations from user inputs.

Flow Chart:

Start

User input for distance, miles per gallon and price per gallon

Cost = (Distance/MPG) \* PPG

Output for Cost

End

**Check List:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N** | **Comments** |
|  | **Source java files** |  |  |
|  | **Compressed files:** |  |  |
|  | FirstInitialLastName\_Project1\_Moss.zip |  |  |
|  | FirstInitialLastName\_Project1\_doc.zip |  |  |
|  | **Program compiles** |  |  |
|  | **Program runs with desired outputs related to a Test Plan** |  |  |
|  | **Checklist is completed and included in the Documentation** |  |  |
|  | **Documentation file:** |  |  |
|  | **Comprehensive Test Plan** |  |  |
|  | **Screenshots based on Test Plan** |  |  |
|  | **UML Diagram** |  |  |
|  | **Algorithms/Pseudocode** |  |  |
|  | **FlowChart** |  |  |
|  | **Lessons Learned** |  |  |