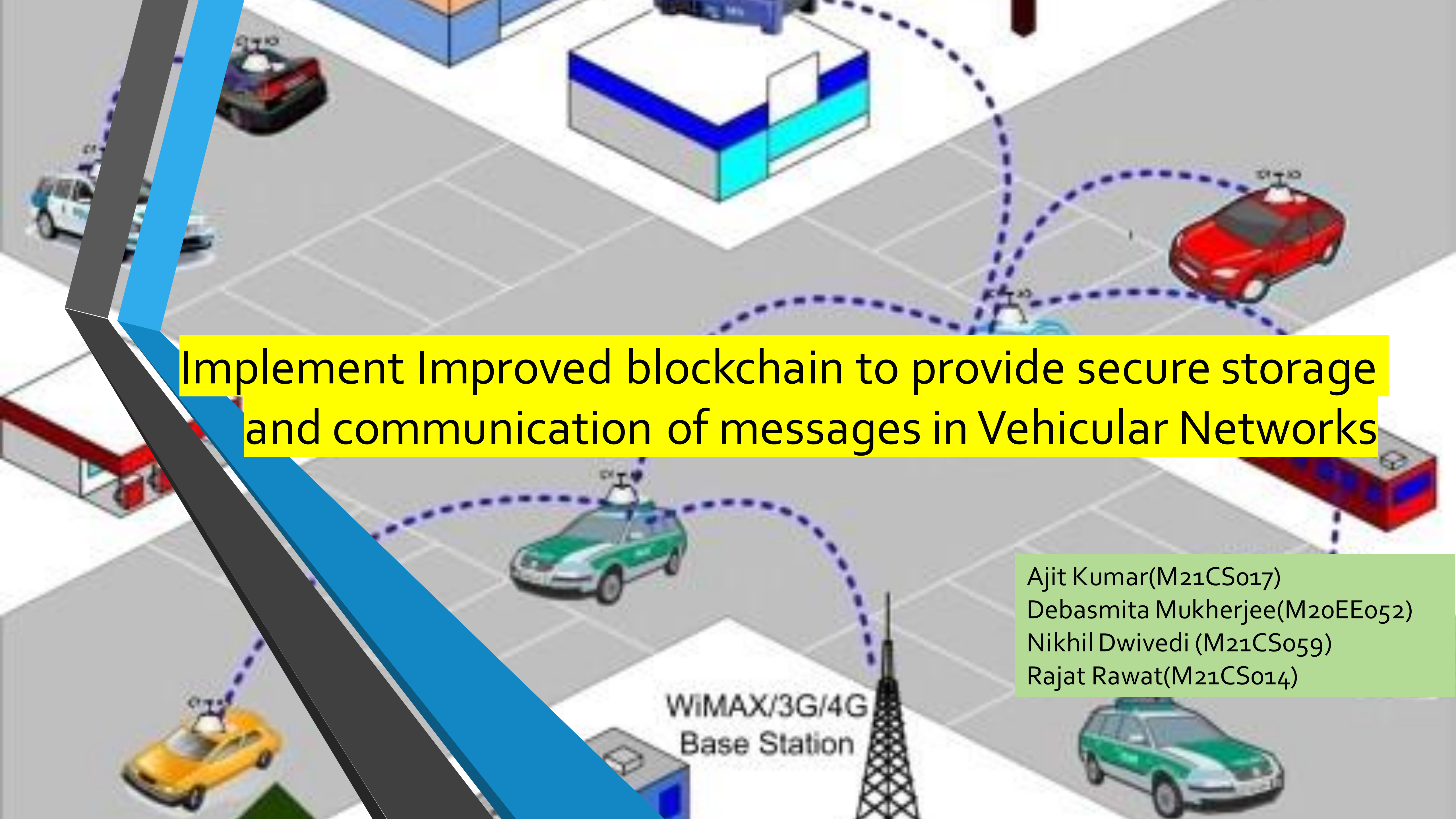
Implement Improved blockchain to provide secure storage and communication of messages in Vehicular Networks

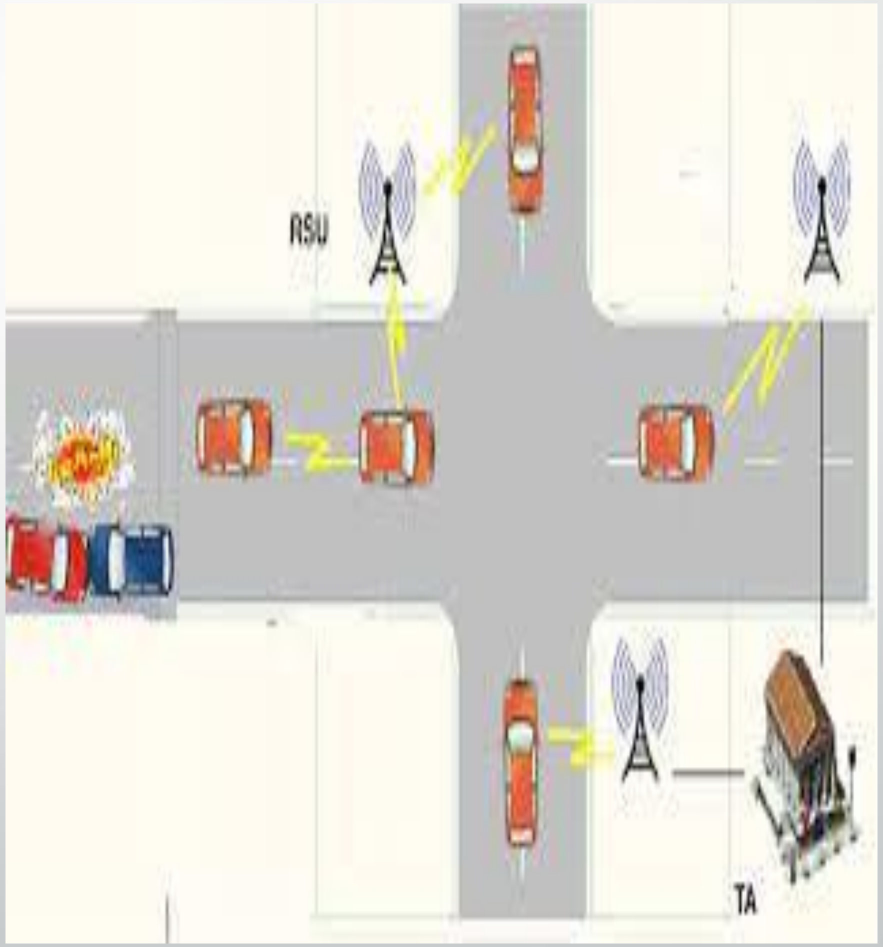
Ajit Kumar(M21CS017)
Debasmita Mukherjee(M20EE052)
Nikhil Dwivedi (M21CS059)
Rajat Rawat(M21CS014)



Introduction

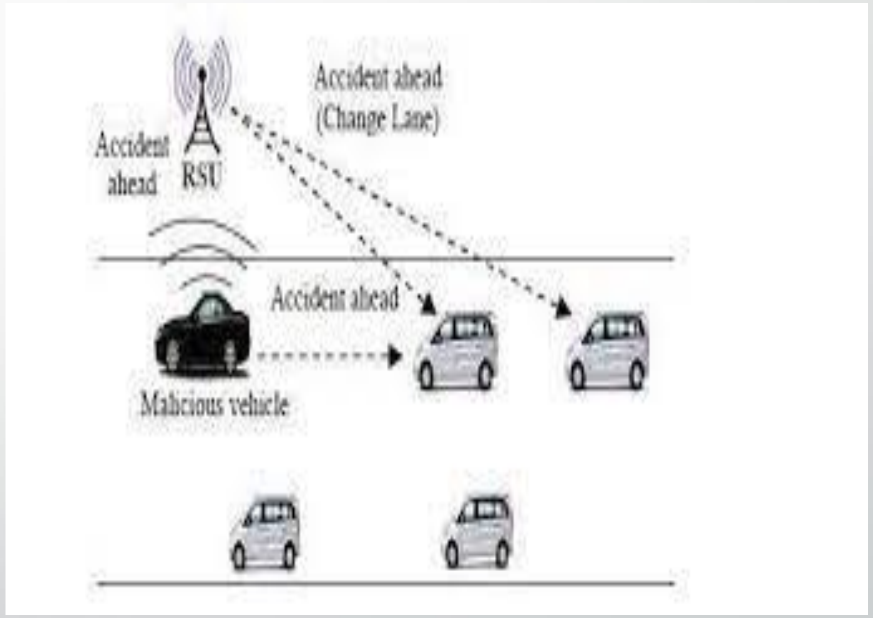
- Recently increase in vehicles leads to:
  - Increase in accidents
  - Traffic jams
  - Wastage of time

- What we require to do:
  - Reduce these life threatening events
  - Securely channelizing the info to peer vehicles
  - Overall reduce accidents and time



Problem Statement

- Provide secure communication of messages among vehicles
- More focus on trustworthiness of messages, as:
  - Malicious vehicles can be present
  - They provide false info
  - Imp messages cannot be send accurately in real time



- Provide all this info in dynamic VANET environment and in presence of malicious vehicles

Objective

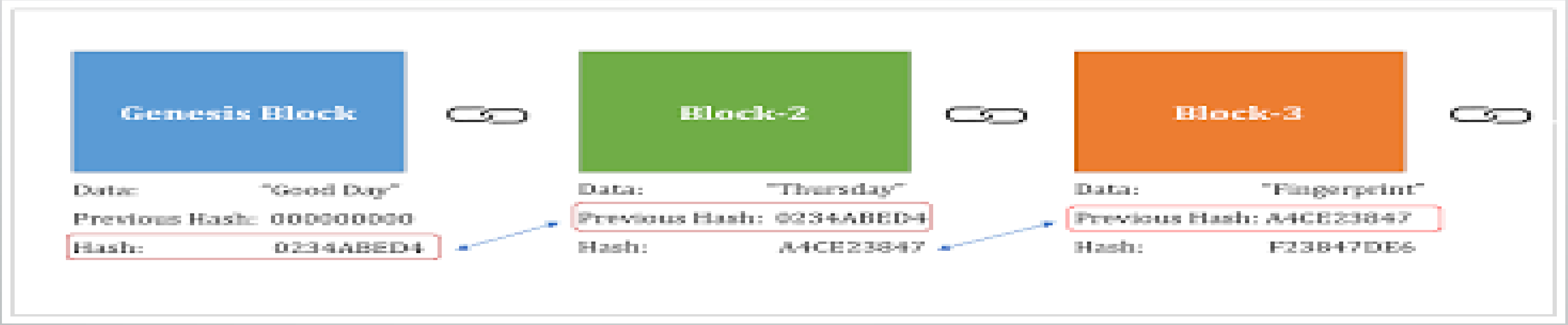
- To reduce life threatening events in dynamic VANET environment along with malicious nodes
- Create a blockchain for message exchange among vehicles within a country

Improvised Blockchain in VANET

Traditional Blockchain	Our Proposed Blockchain
Deal with cryptocurrency & Transactions	Deals with Vehicles & Safety messages
Maintain info of all nodes/ users in world	Need not connect countries which are geographically not connected

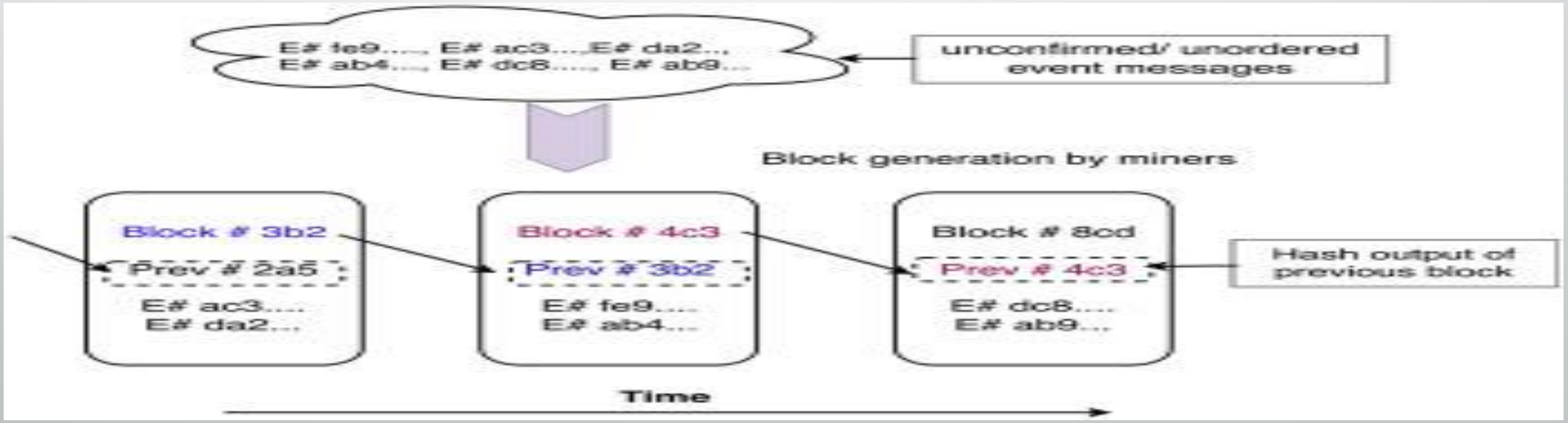
What exactly in a Block

- Starting block called Genesis block
- Each block has:
  - Data
  - Its own hash
  - Hash of previous block, through which they are linked
  - Timestamp at which that block last updated



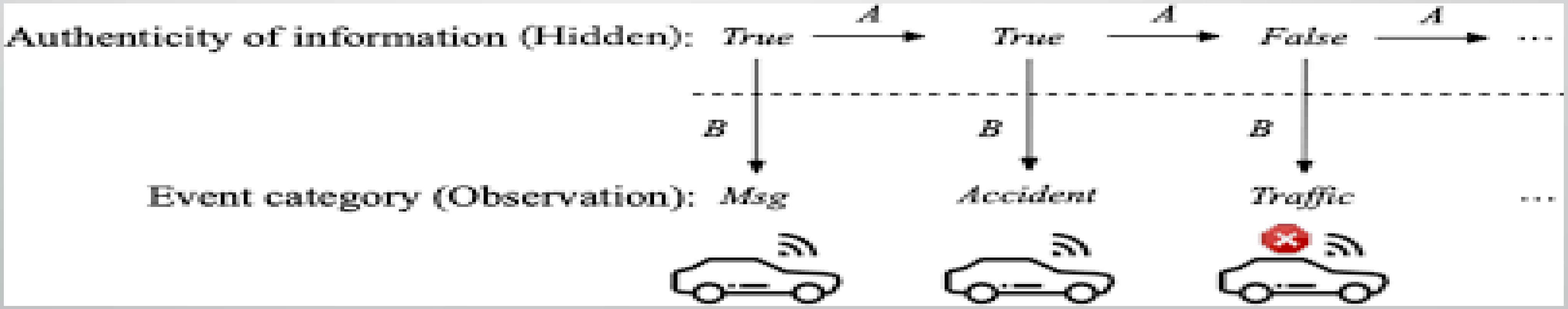
Blockchain Implementation in VANET

- Blockchain:
  - Chain of blocks
  - Each block knows the hash of previous block
  - Hashes of all blocks are chained together in sequential (linked list) fashion to build a blockchain
  - Hash of a block calculated by aggregating the contents of that block


E# fe9....., E# ac3....,E# da2...,
E# ab4...., E# dc8....., E# ab9....
unconfirmed/ unordered event messages
Block generation by miners
Block # 3b2
Prev # 2a5
E# ac3....
E# da2....
Block # 4c3
Prev # 3b2
E# fe9....
E# ab4....
Block # 8cd
Prev # 4c3
E# dc8....
E# ab9...
Hash output of previous block
Time

How to know the trustworthiness of a message

- Sender sends message
- In range vehicles transmit further depending on trueness of sender vehicle
- If information is correct:
  - Truenss of Sender Incremented -> Message transmitted to other vehicles
- Else if information is wrong:
  - Trueness of Sender Decremented -> Message not transmitted

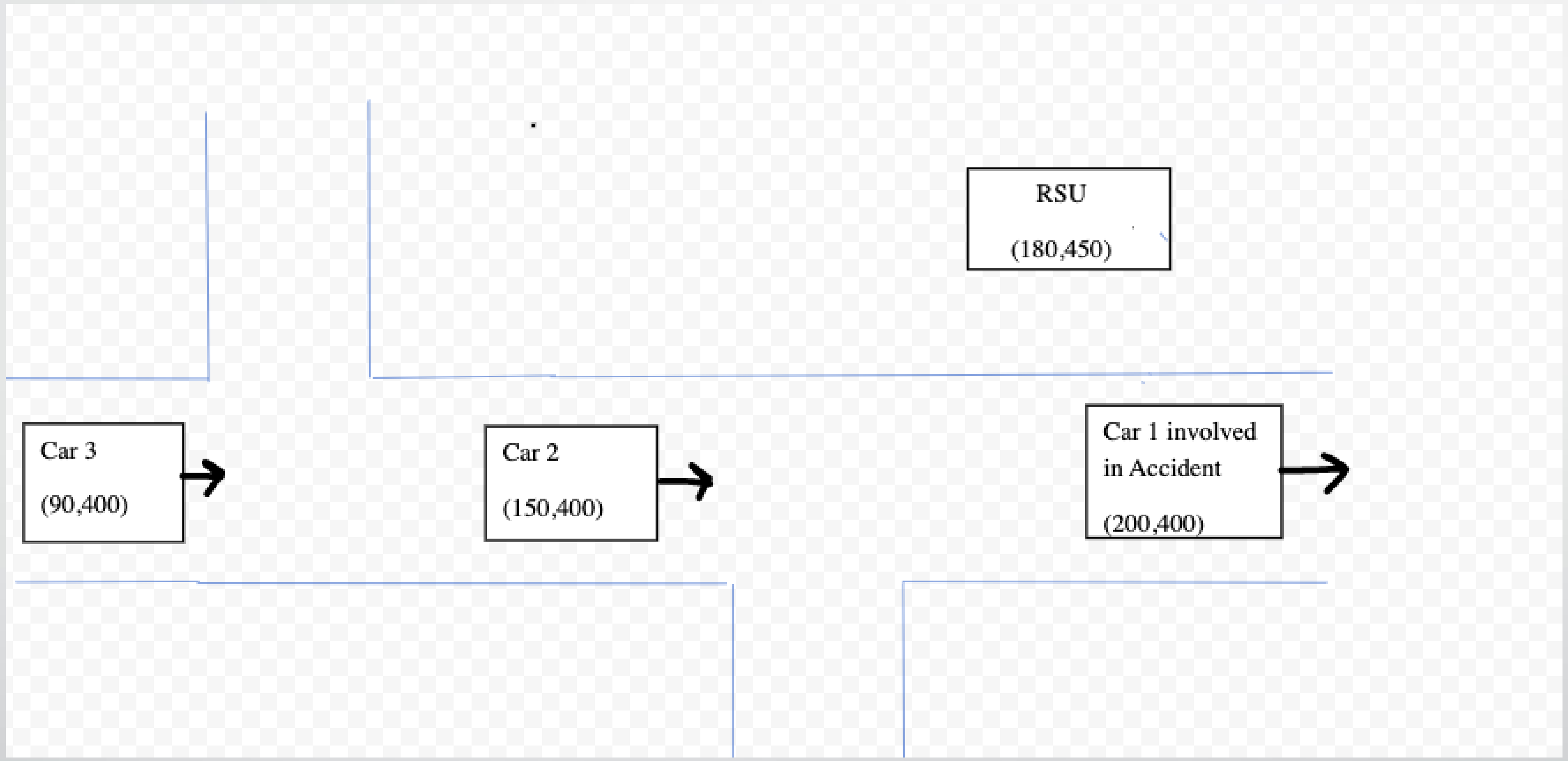


Authenticity of information (Hidden):
True
A
True
A
False
A
...
B
B
B
Event category (Observation):
Msg
Accident
Traffic
...

Tools used for Coding

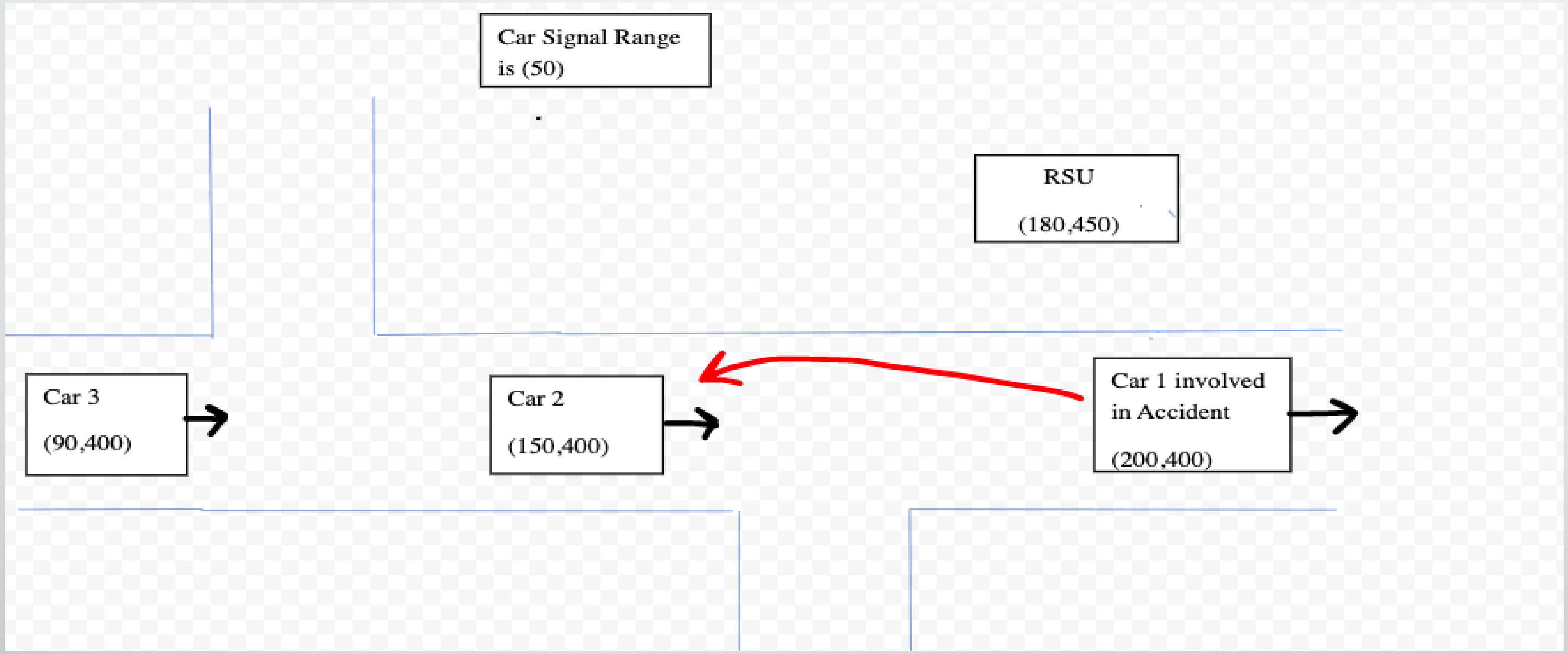
- Java Language for Implementation
- Object Oriented Programming Concepts

Scenario Explaination-1



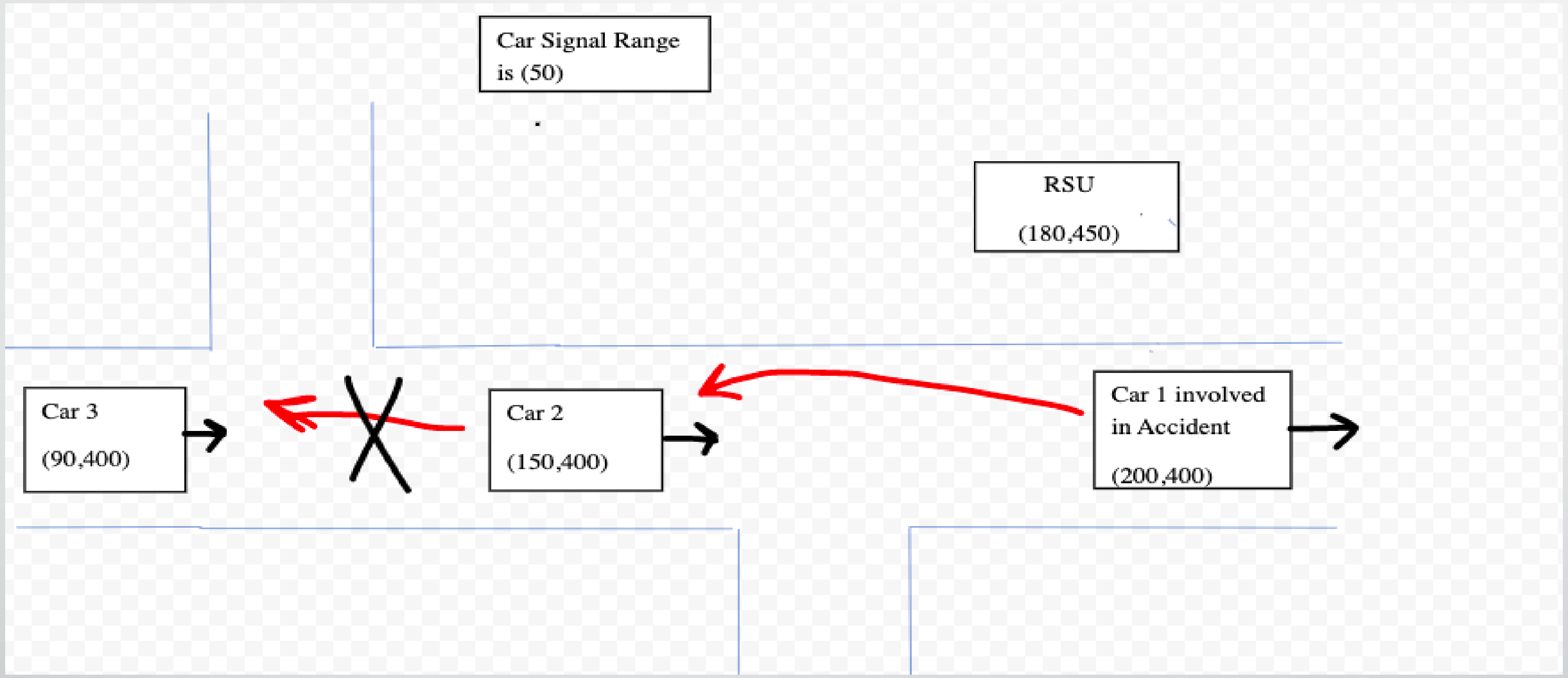
Initial Scenario with Car 1 involved in accident

Scenario Explaination-2



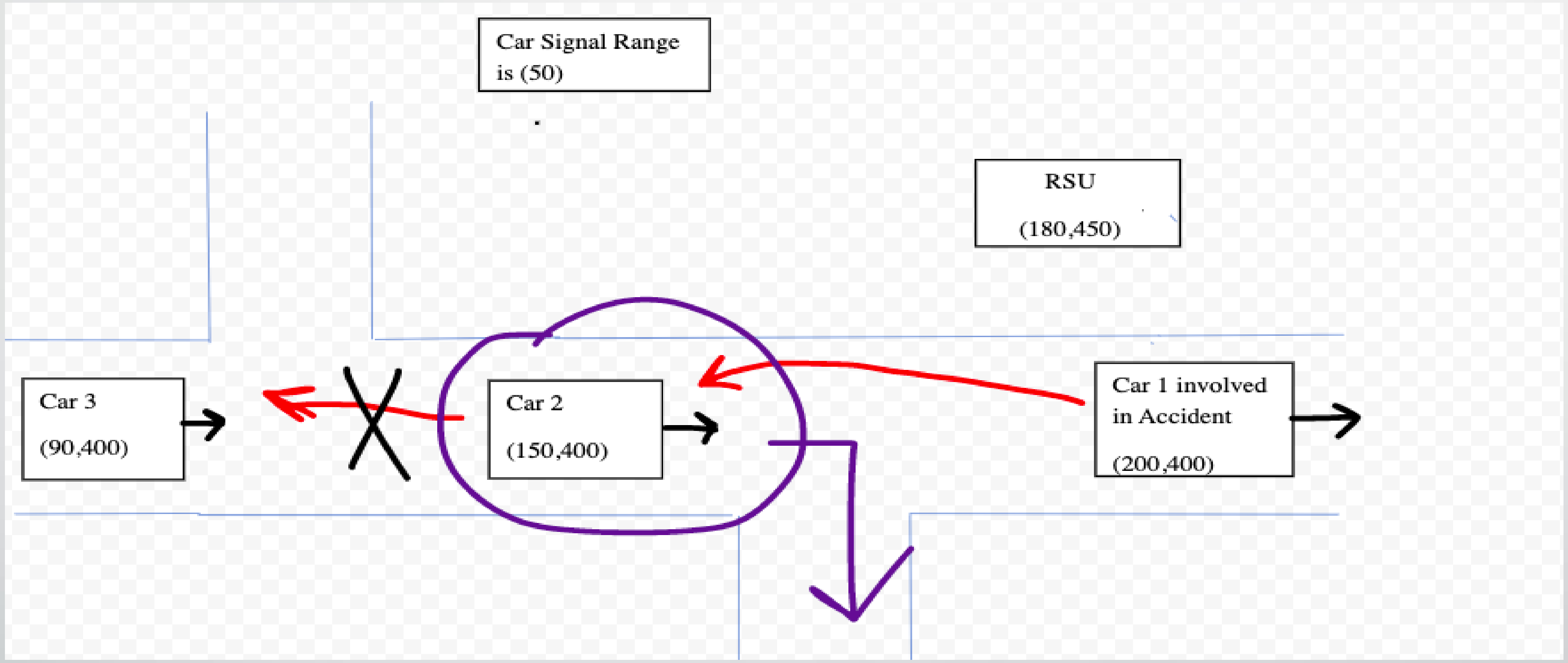
Car 1 sends V2V signal to Car 2 only, as Car 2 is in range (50) of Car 1

Scenario Explaination-3



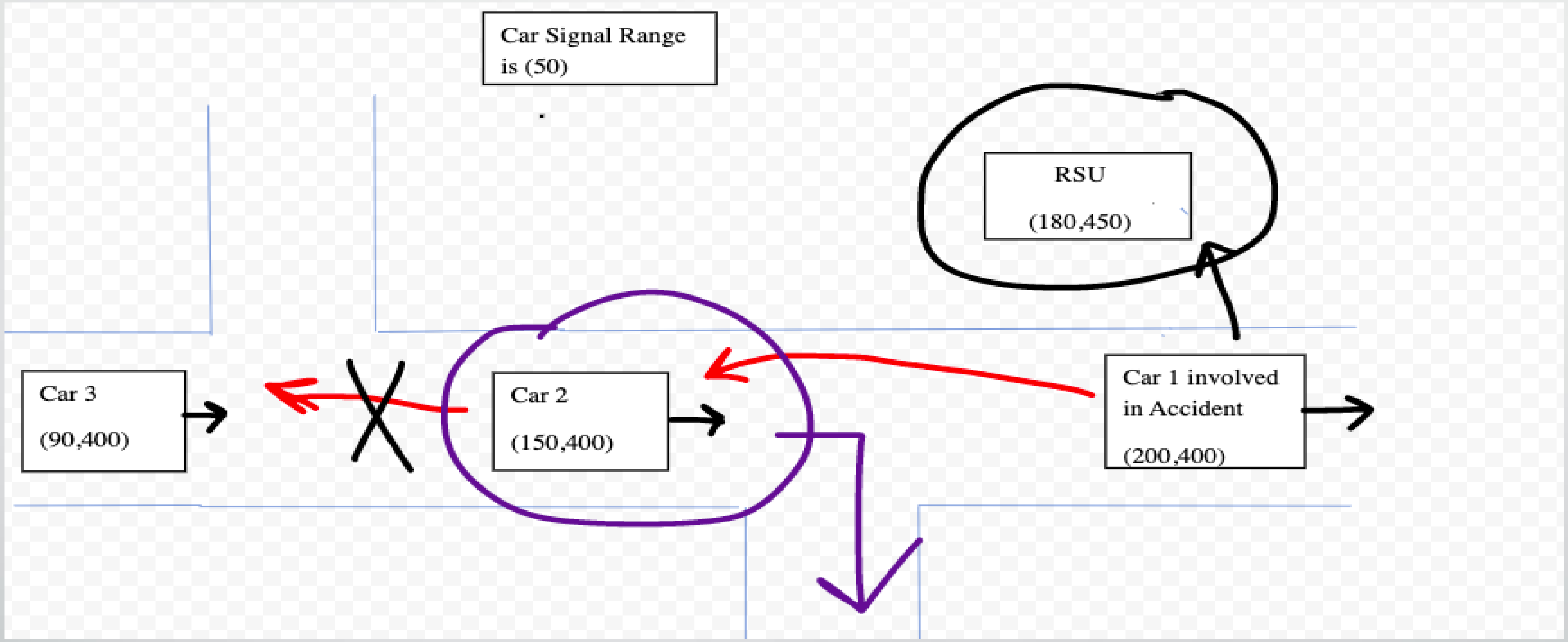
Car 2 cannot send V2V signal to Car 3, as dis b/w car2 and car3 >50

Scenario Explaination-4



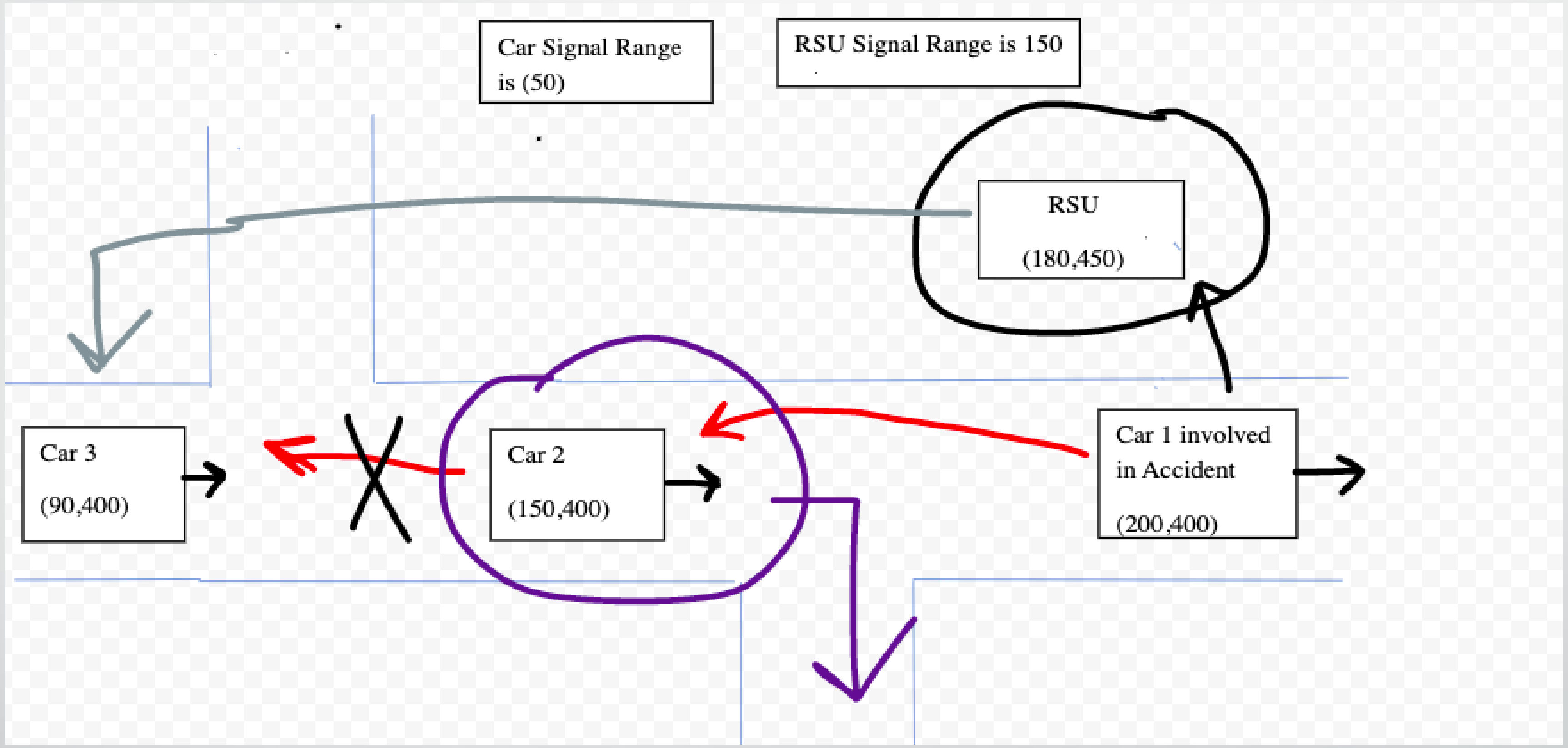
Car 3 cannot send to anyone & Car 2 changes direction to avoid congestion

Scenario Explaination-5



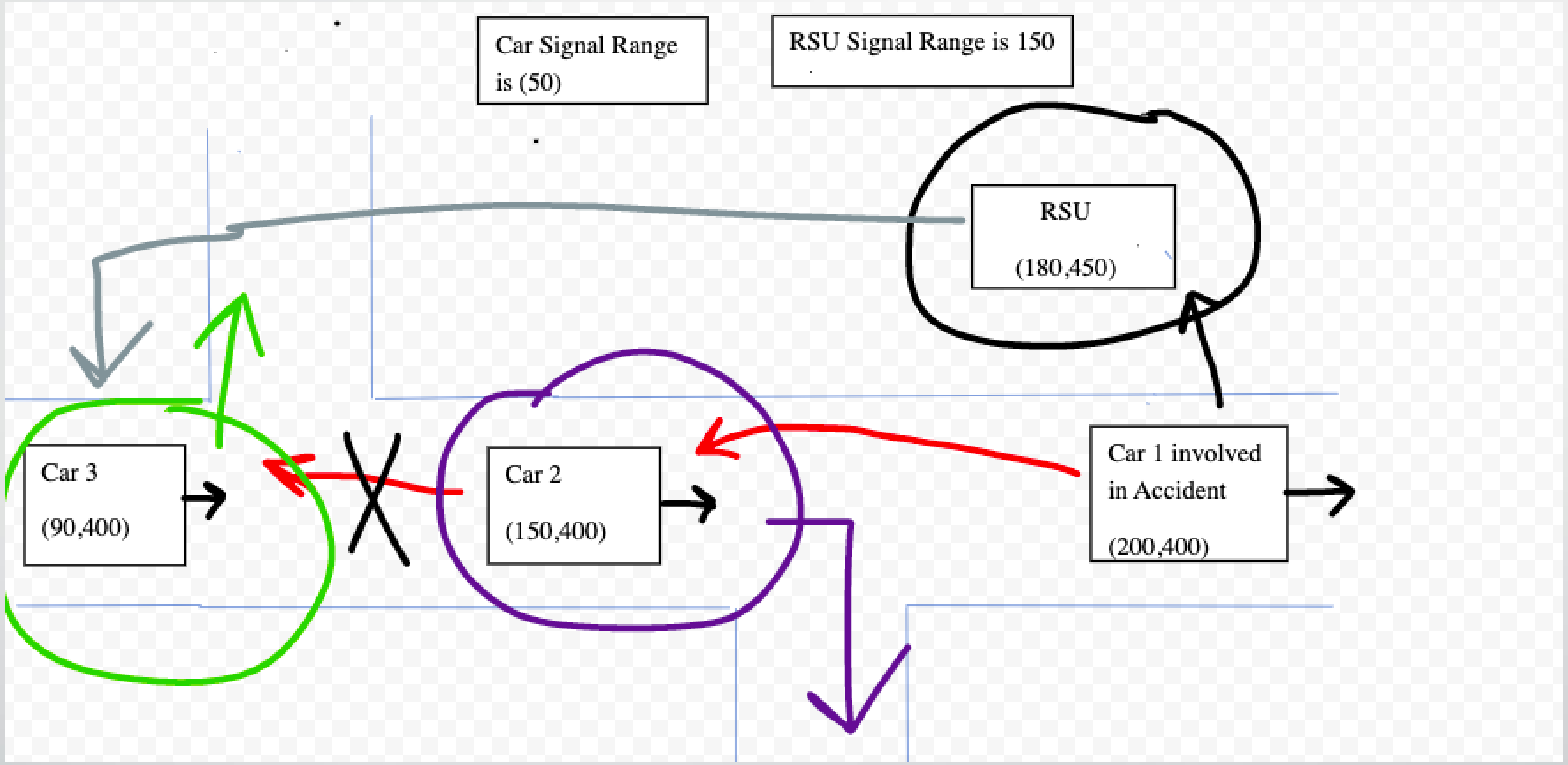
Car 1 sends to RSU, as it is inside its range

Scenario Explaination-6



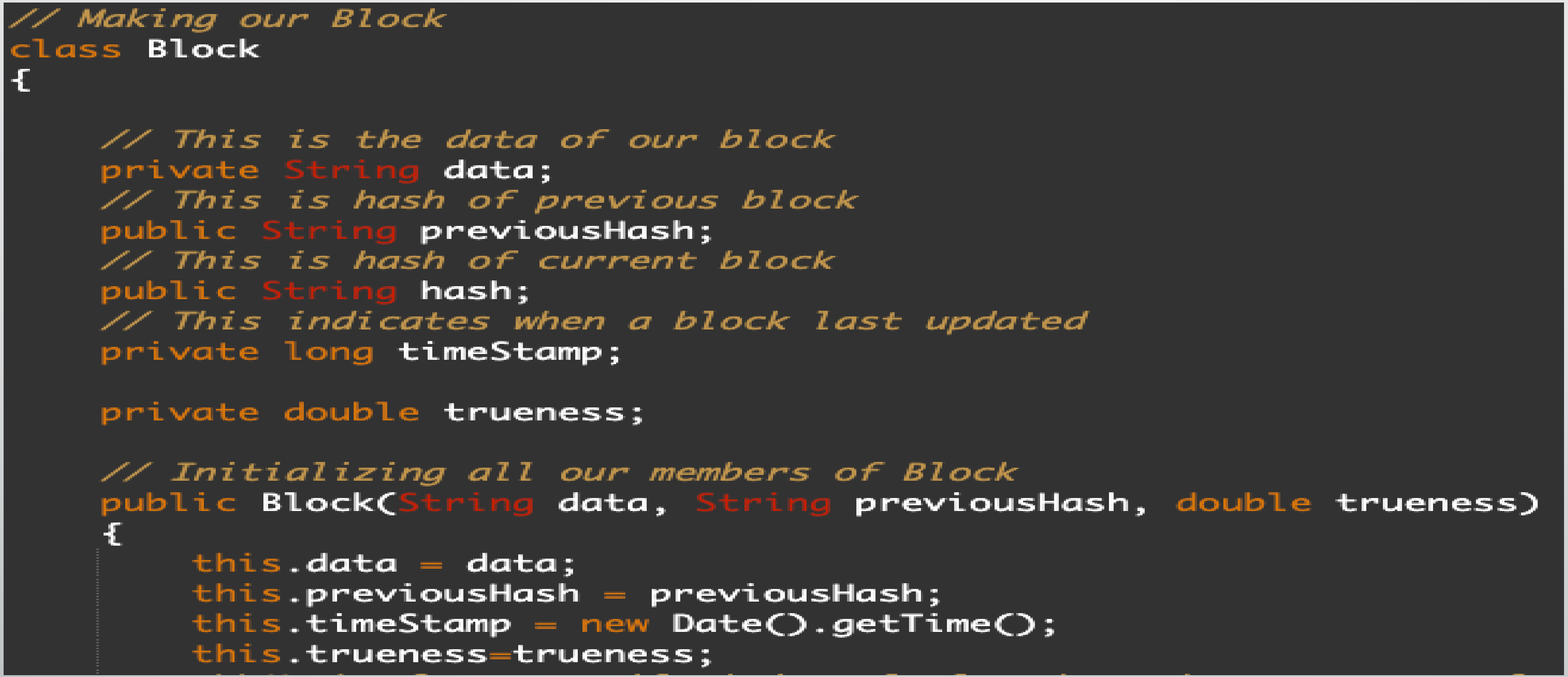
RSU sends to Car 2 & Car 3, but Car2 already got V2V from Car 1, so it avoid

Scenario Explaination-7



Car 3 changes its direction to avoid congestion

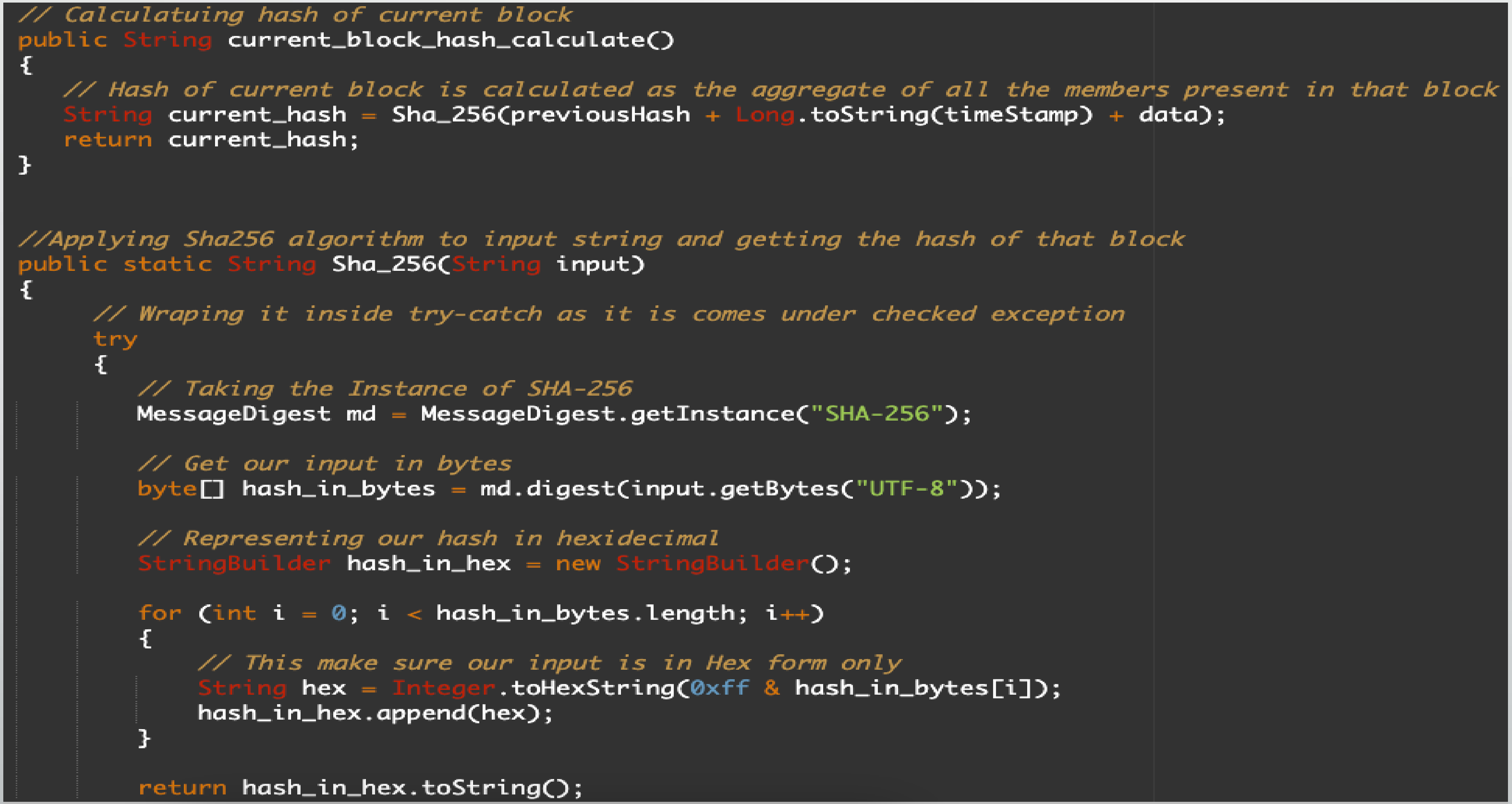
Code Implementation-1



```
// Making our Block
class Block
{

    // This is the data of our block
    private String data;
    // This is hash of previous block
    public String previousHash;
    // This is hash of current block
    public String hash;
    // This indicates when a block last updated
    private long timeStamp;

    private double trueness;

    // Initializing all our members of Block
    public Block(String data, String previousHash, double trueness)
    {
        this.data = data;
        this.previousHash = previousHash;
        this.timeStamp = new Date().getTime();
        this.trueness=trueness;
```

Making Chain of Blocks representing Vehicles
Trueness indicates trust level of each Vehicle

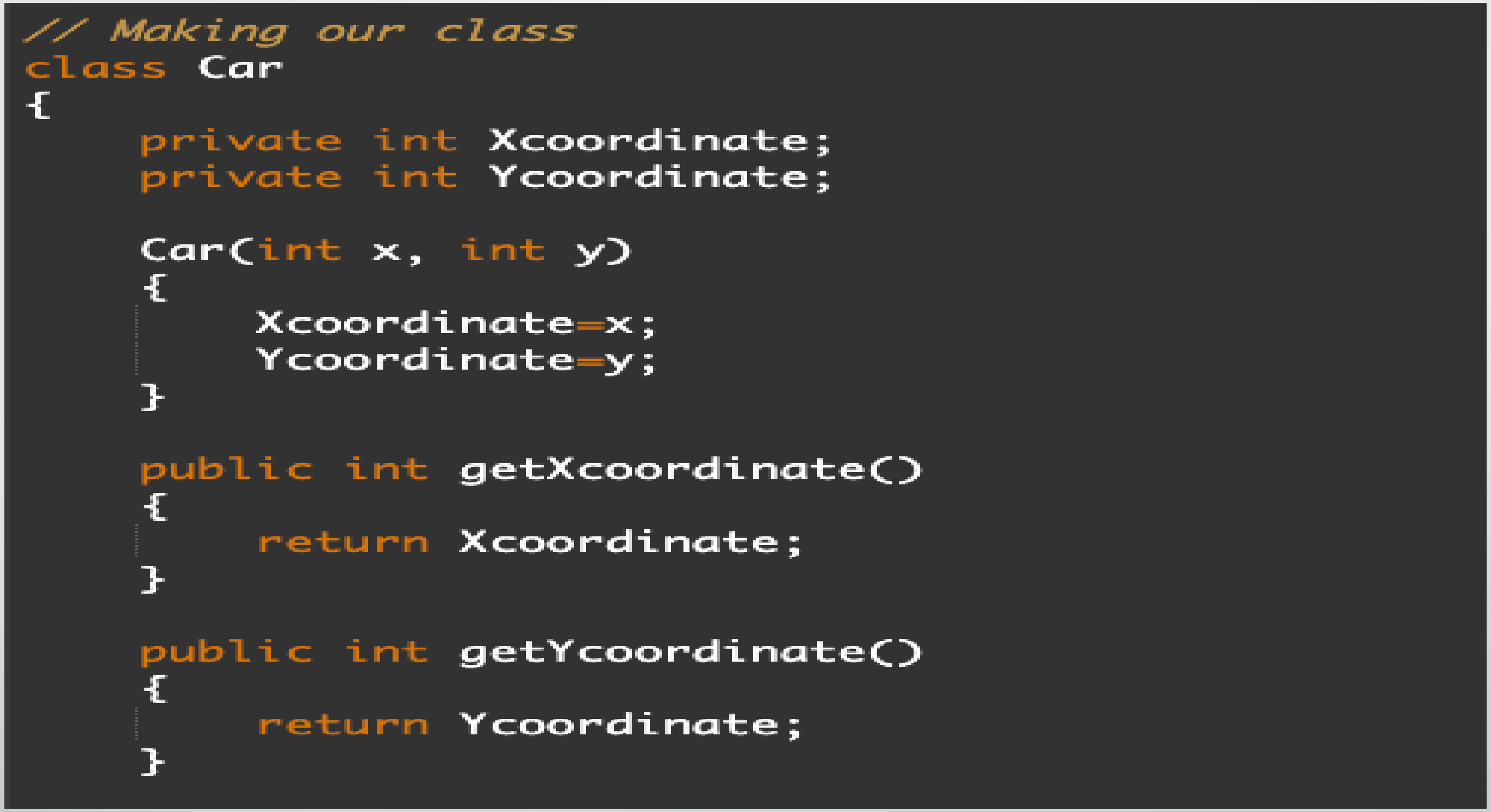
Code Implementation-2



```
// Calculatuing hash of current block
public String current_block_hash_calculate()
{
    // Hash of current block is calculated as the aggregate of all the members present in that block
    String current_hash = Sha_256(previousHash + Long.toString(timeStamp) + data);
    return current_hash;
}


//Applying Sha256 algorithm to input string and getting the hash of that block
public static String Sha_256(String input)
{
      // Wraping it inside try-catch as it is comes under checked exception
      try
      {
        // Taking the Instance of SHA-256
        MessageDigest md = MessageDigest.getInstance("SHA-256");

        // Get our input in bytes
        byte[] hash_in_bytes = md.digest(input.getBytes("UTF-8"));

        // Representing our hash in hexidecimal
        StringBuilder hash_in_hex = new StringBuilder();

        for (int i = 0; i < hash_in_bytes.length; i++)
        {
            // This make sure our input is in Hex form only
            String hex = Integer.toHexString(0xff & hash_in_bytes[i]);
            hash_in_hex.append(hex);
        }

        return hash_in_hex.toString();
```

Finding Hash of each block using SHA-256

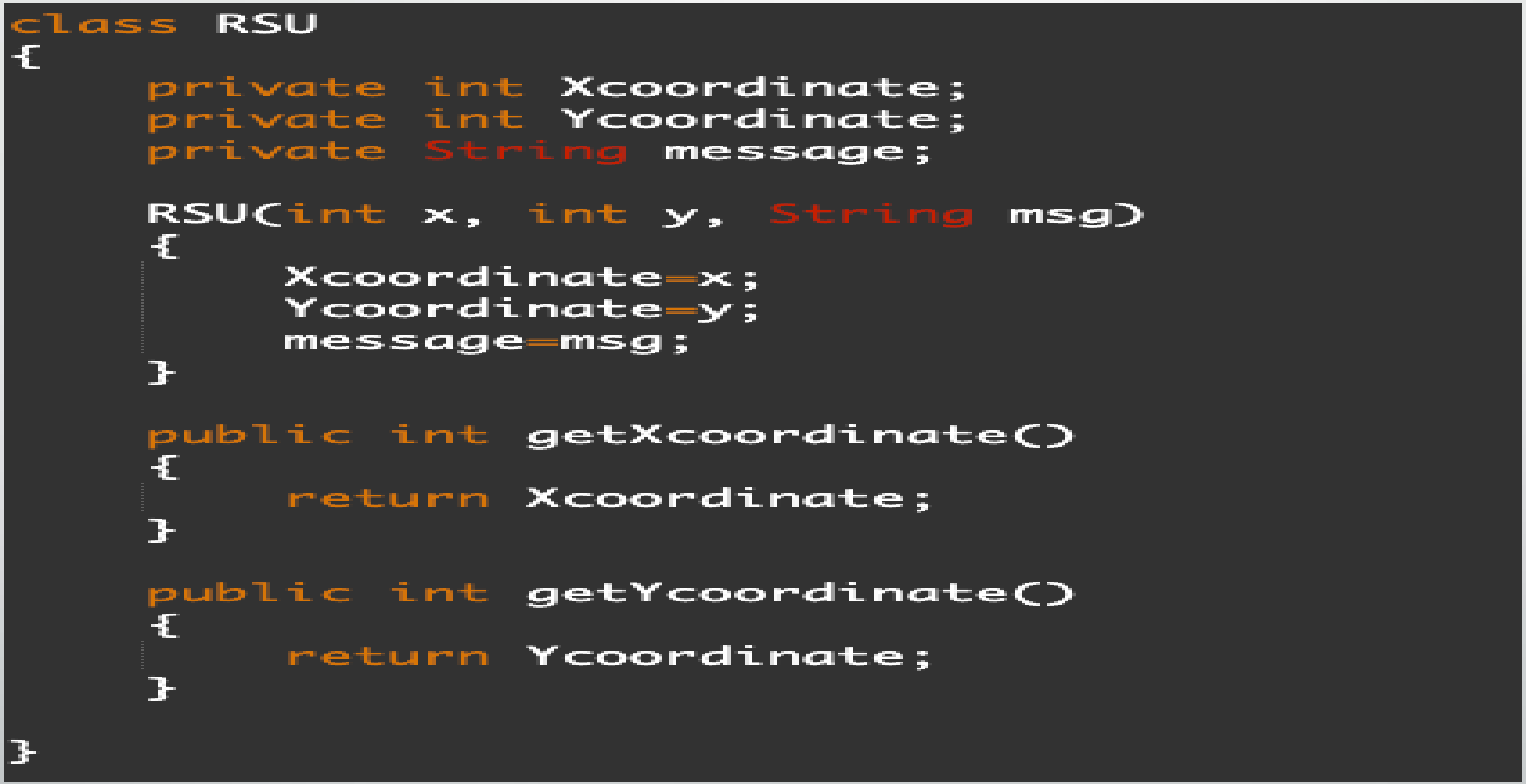
Code Implementation-3



```
// Making our class
class Car
{
    private int Xcoordinate;
    private int Ycoordinate;

    Car(int x, int y)
    {
        Xcoordinate=x;
        Ycoordinate=y;
    }

    public int getXcoordinate()
    {
        return Xcoordinate;
    }

    public int getYcoordinate()
    {
        return Ycoordinate;
    }
```

Car class to keep track of coordinates of Car

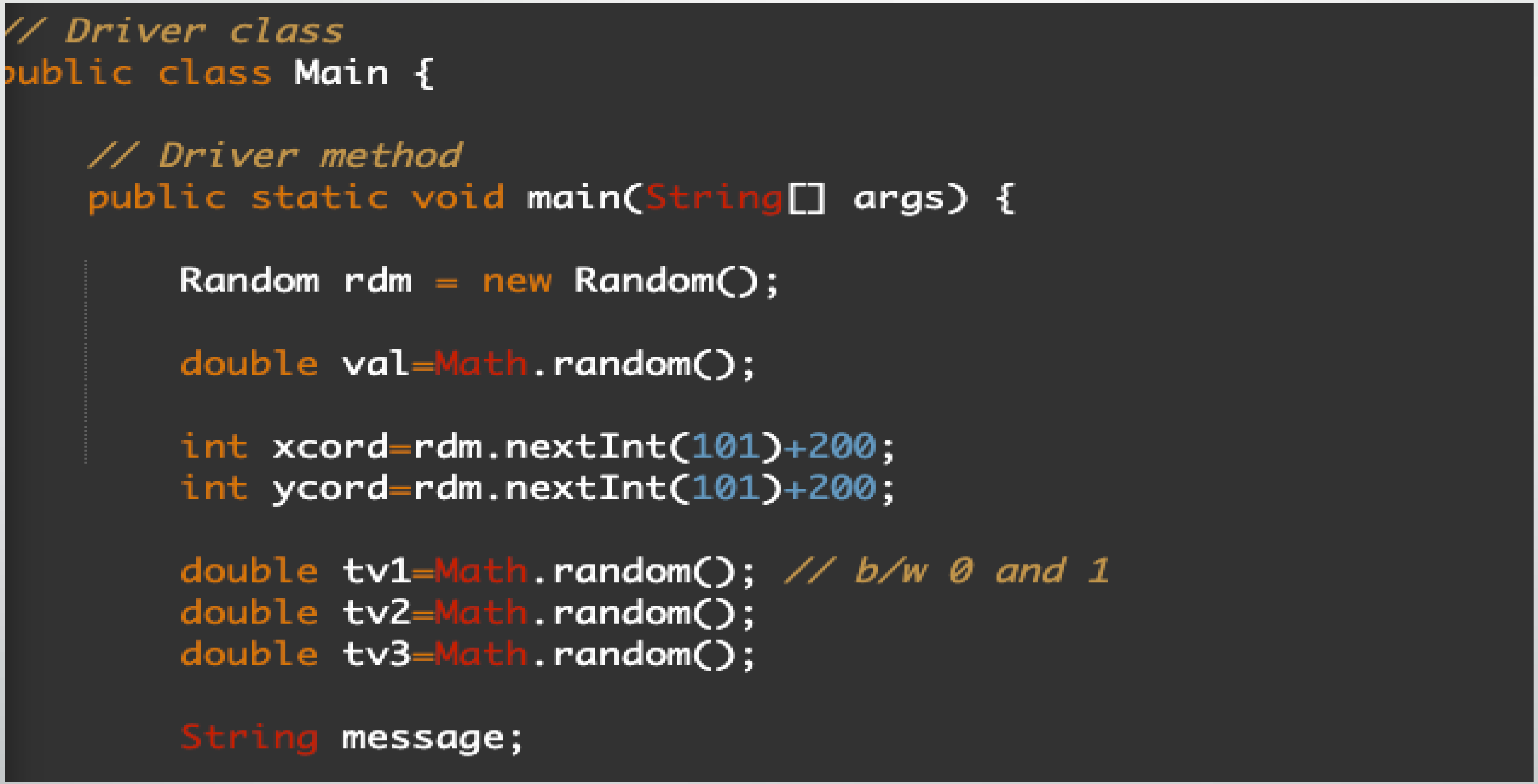
Code Implementation-4



```
class RSU
{
        private int Xcoordinate;
        private int Ycoordinate;
        private String message;

        RSU(int x, int y, String msg)
        {
                Xcoordinate=x;
                Ycoordinate=y;
                message=msg;
        }

        public int getXcoordinate()
        {
                return Xcoordinate;
        }

        public int getYcoordinate()
        {
                return Ycoordinate;
        }

}
```

RSU class to keep track of coordinates of RSU

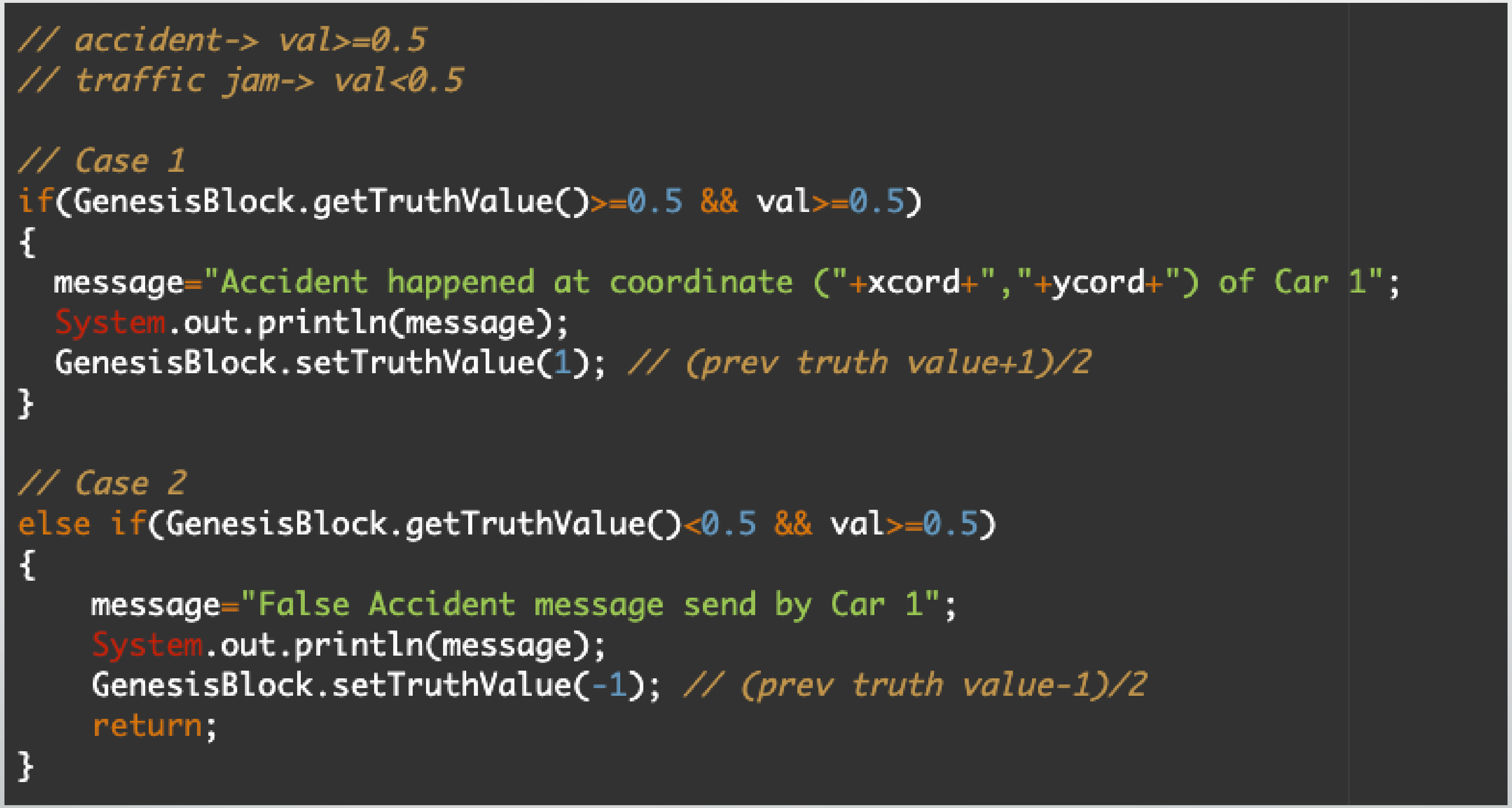
Code Implementation-5



```
// Driver class
ublic class Main {

    // Driver method
    public static void main(String[] args) {

        Random rdm = new Random();

        double val=Math.random();

        int xcord=rdm.nextInt(101)+200;
        int ycord=rdm.nextInt(101)+200;

        double tv1=Math.random(); // b/w 0 and 1
        double tv2=Math.random();
        double tv3=Math.random();

        String message;
```

Taking Random coordinates & Random Trueness value of each 3 Cars

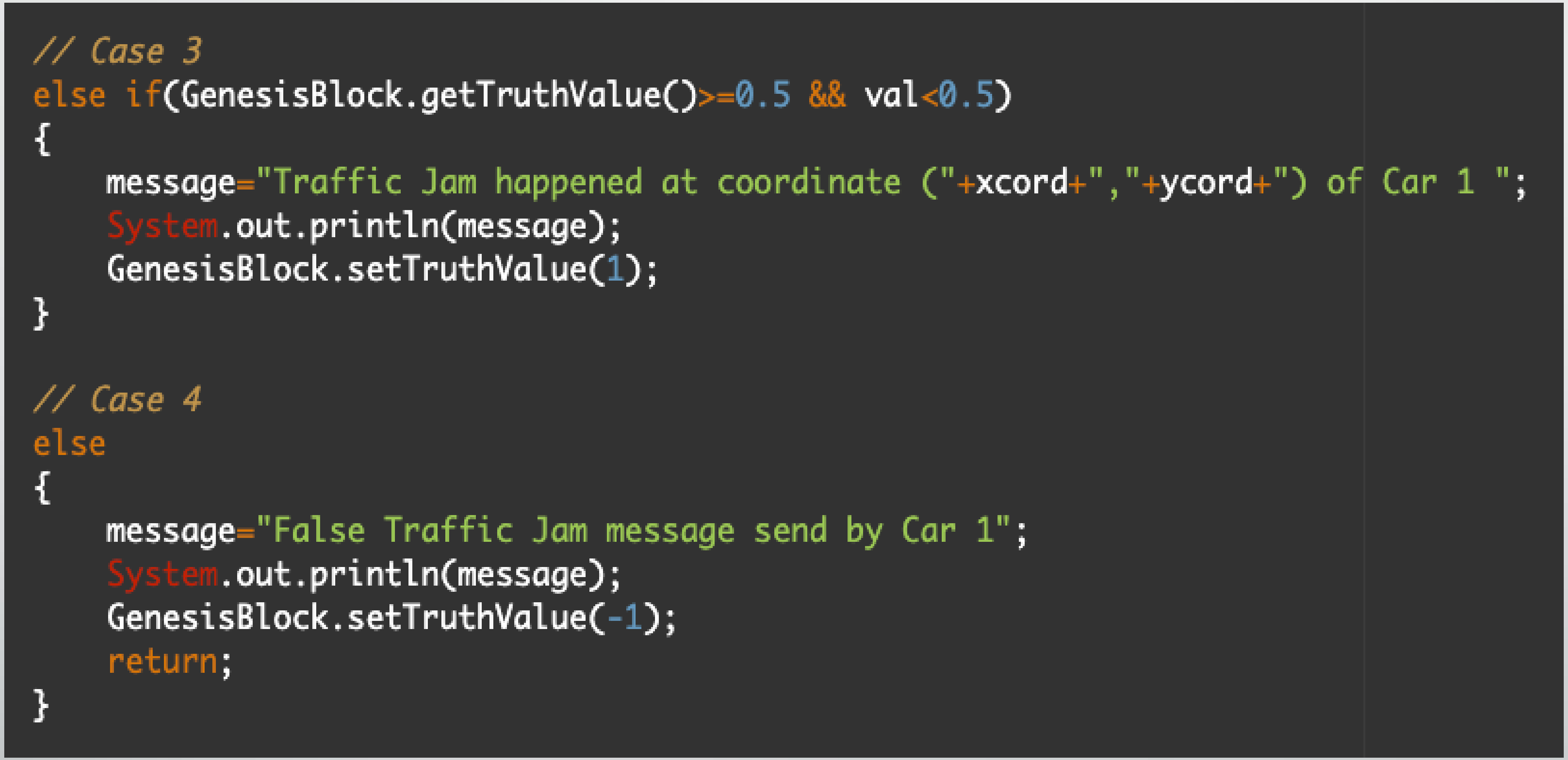
Code Implementation-6



```
// accident-> val>=0.5
// traffic jam-> val<0.5

// Case 1
if(GenesisBlock.getTruthValue()>=0.5 && val>=0.5)
{
  message="Accident happened at coordinate ("+xcord+","+ycord+") of Car 1";
  System.out.println(message);
  GenesisBlock.setTruthValue(1); // (prev truth value+1)/2
}

// Case 2
else if(GenesisBlock.getTruthValue()<0.5 && val>=0.5)
{
    message="False Accident message send by Car 1";
    System.out.println(message);
    GenesisBlock.setTruthValue(-1); // (prev truth value-1)/2
    return;
}
```

Case 1: Accident happened & information is True (+1 trueness)
Case 2: Accident happened but it's a False info transmitted (-1 trueness)

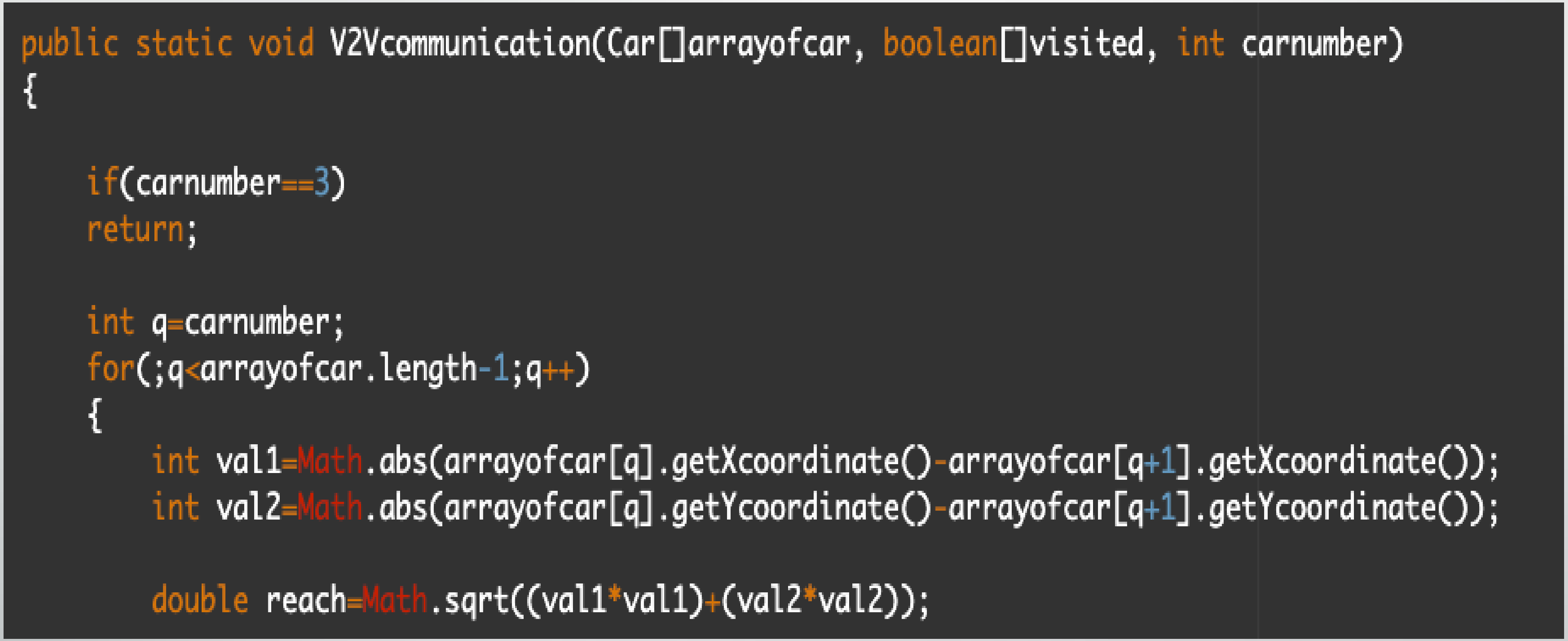
Code Implementation-7



```
// Case 3
else if(GenesisBlock.getTruthValue()>=0.5 && val<0.5)
{
    message="Traffic Jam happened at coordinate ("+xcord+","+ycord+") of Car 1 ";
    System.out.println(message);
    GenesisBlock.setTruthValue(1);
}

// Case 4
else
{
    message="False Traffic Jam message send by Car 1";
    System.out.println(message);
    GenesisBlock.setTruthValue(-1);
    return;
}
```

Case 3: Traffic Jam happened & information is True (+1 trueness)
Case 4: Traffic Jam happened but it’s a False info transmitted (-1 trueness)

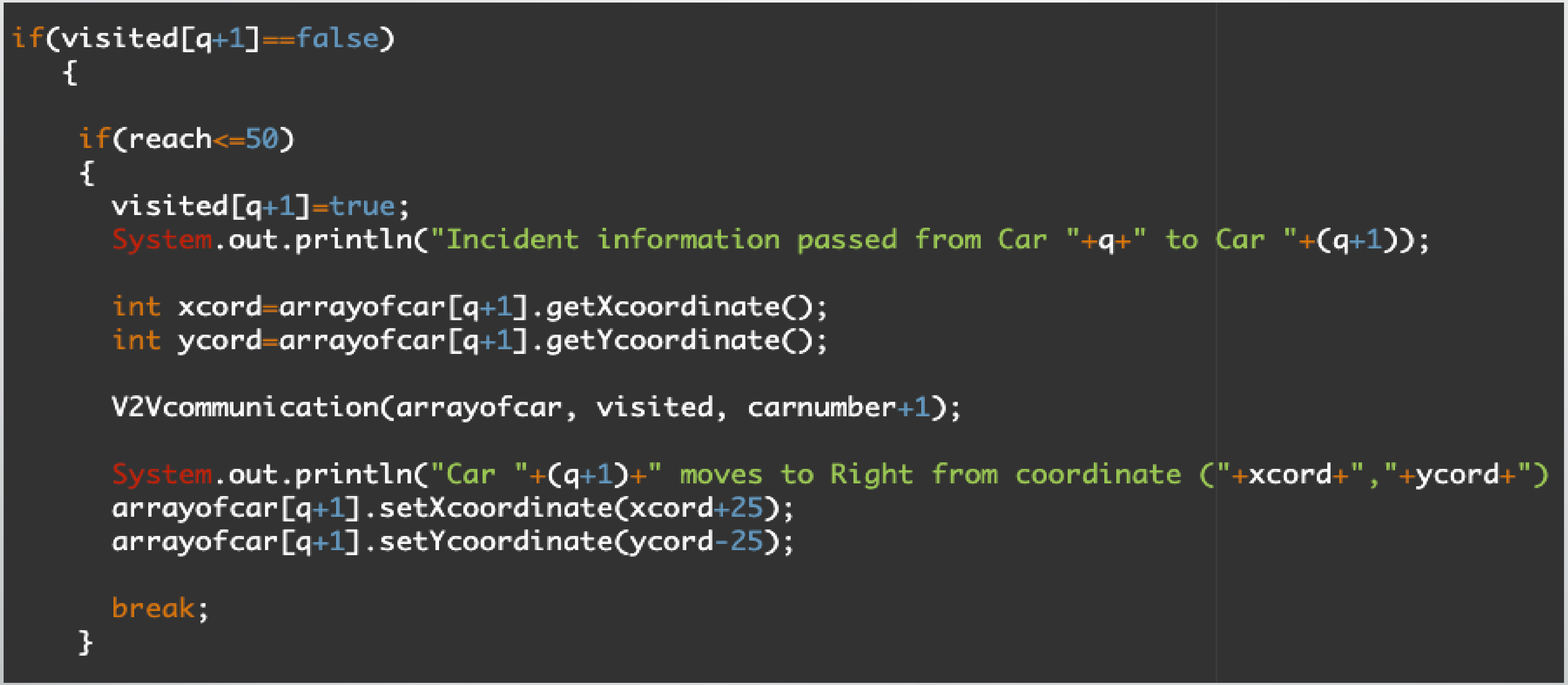
Code Implementation-8



```
public static void V2Vcommunication(Car[]arrayofcar, boolean[]visited, int carnumber)
{

    if(carnumber==3)
    return;

    int q=carnumber;
    for(;q<arrayofcar.length-1;q++)
    {
        int val1=Math.abs(arrayofcar[q].getXcoordinate()-arrayofcar[q+1].getXcoordinate());
        int val2=Math.abs(arrayofcar[q].getYcoordinate()-arrayofcar[q+1].getYcoordinate());

        double reach=Math.sqrt((val1*val1)+(val2*val2));
```

V2V Communication b/w vehicles by finding out which vehicles are in reach of one another using Euclidean Distance, (<=50 (in reach) else not)

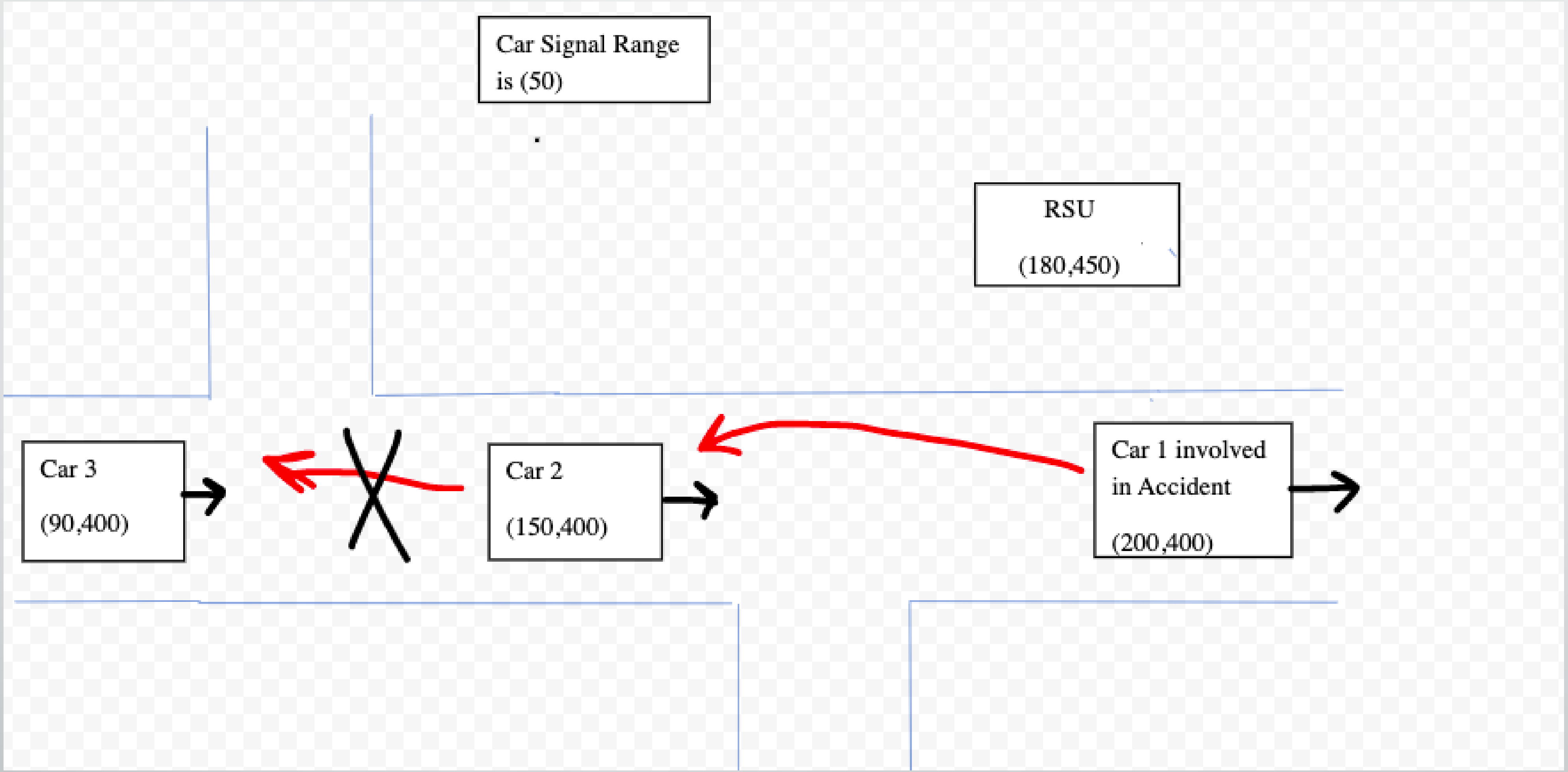
Code Implementation-9



```
if(visited[q+1]==false)
   {

    if(reach<=50)
    {
      visited[q+1]=true;
      System.out.println("Incident information passed from Car "+q+" to Car "+(q+1));

      int xcord=arrayofcar[q+1].getXcoordinate();
      int ycord=arrayofcar[q+1].getYcoordinate();

      V2Vcommunication(arrayofcar, visited, carnumber+1);

      System.out.println("Car "+(q+1)+" moves to Right from coordinate ("+xcord+","+ycord+")
      arrayofcar[q+1].setXcoordinate(xcord+25);
      arrayofcar[q+1].setYcoordinate(ycord-25);

      break;
    }
```

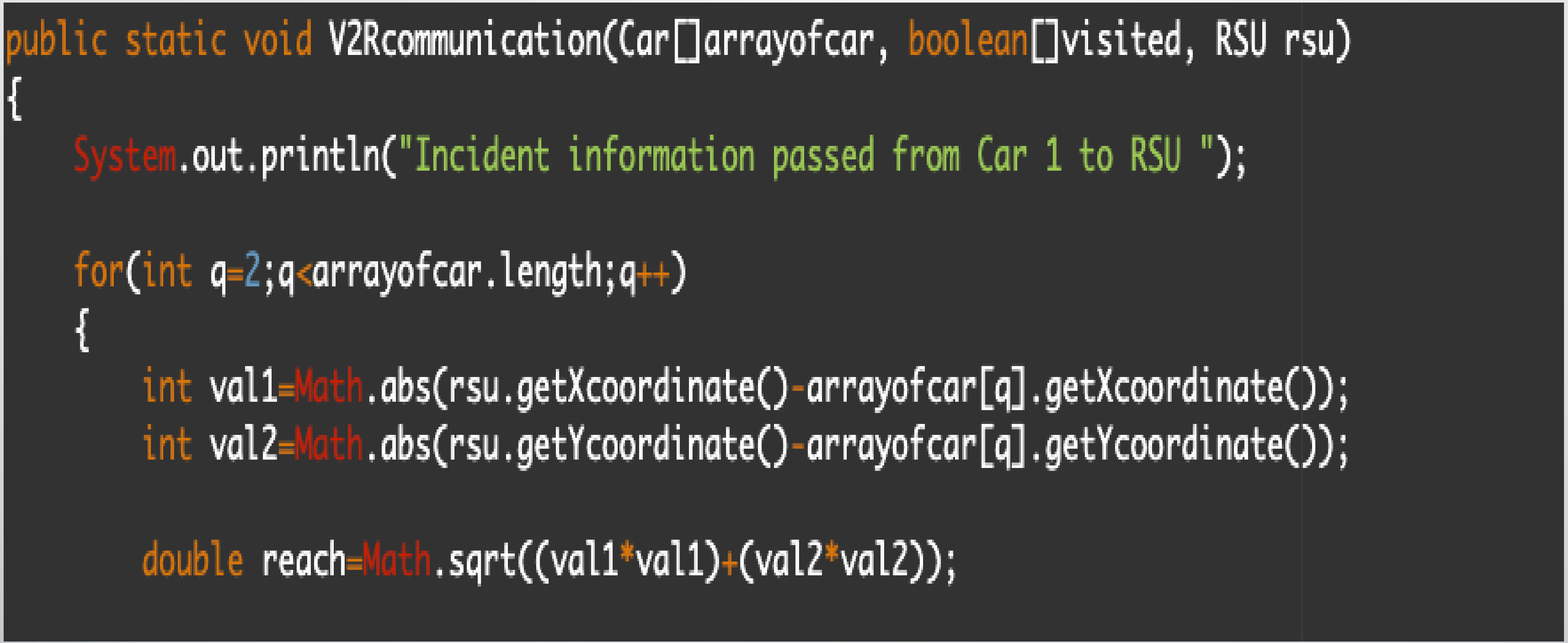
Sending V2V signal only if dis<=50 & that vehicle is not yet explored

Code Implementation-9



Car 1 send to Car 2 but not to Car 3, similarly Car 2 cannot send to Car 3
Also, Car 2 cannot send to Car 1 since it is already explored (avoid loop)

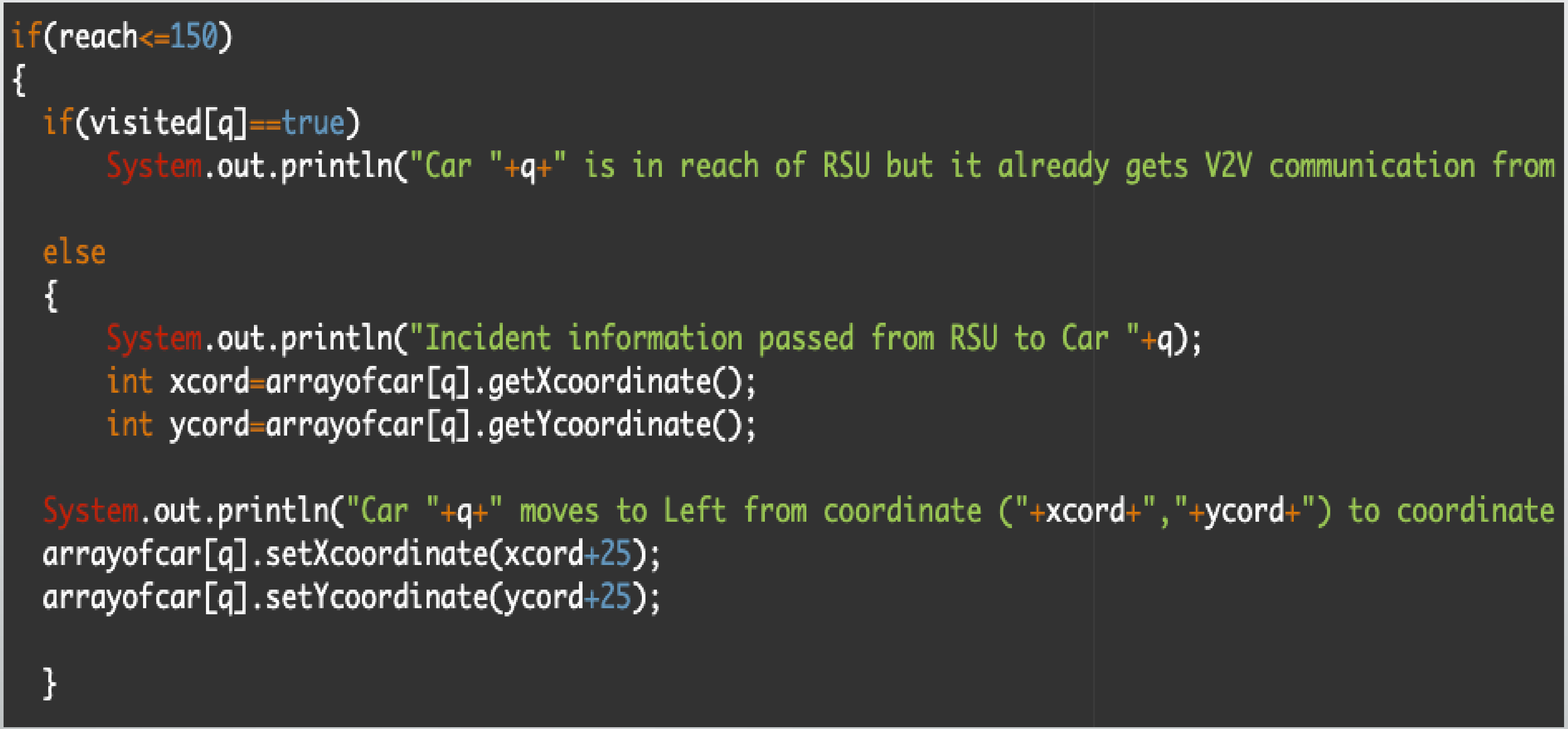
Code Implementation-10



```
public static void V2Rcommunication(Car[]arrayofcar, boolean[]visited, RSU rsu)
{
    System.out.println("Incident information passed from Car 1 to RSU ");

    for(int q=2;q<arrayofcar.length;q++)
    {
        int val1=Math.abs(rsu.getXcoordinate()-arrayofcar[q].getXcoordinate());
        int val2=Math.abs(rsu.getYcoordinate()-arrayofcar[q].getYcoordinate());

        double reach=Math.sqrt((val1*val1)+(val2*val2));
```

V2R Communication b/w vehicle & RSU by finding out which vehicles are in reach of RSU by using Euclidean Distance, (<=150 (in reach) else not)

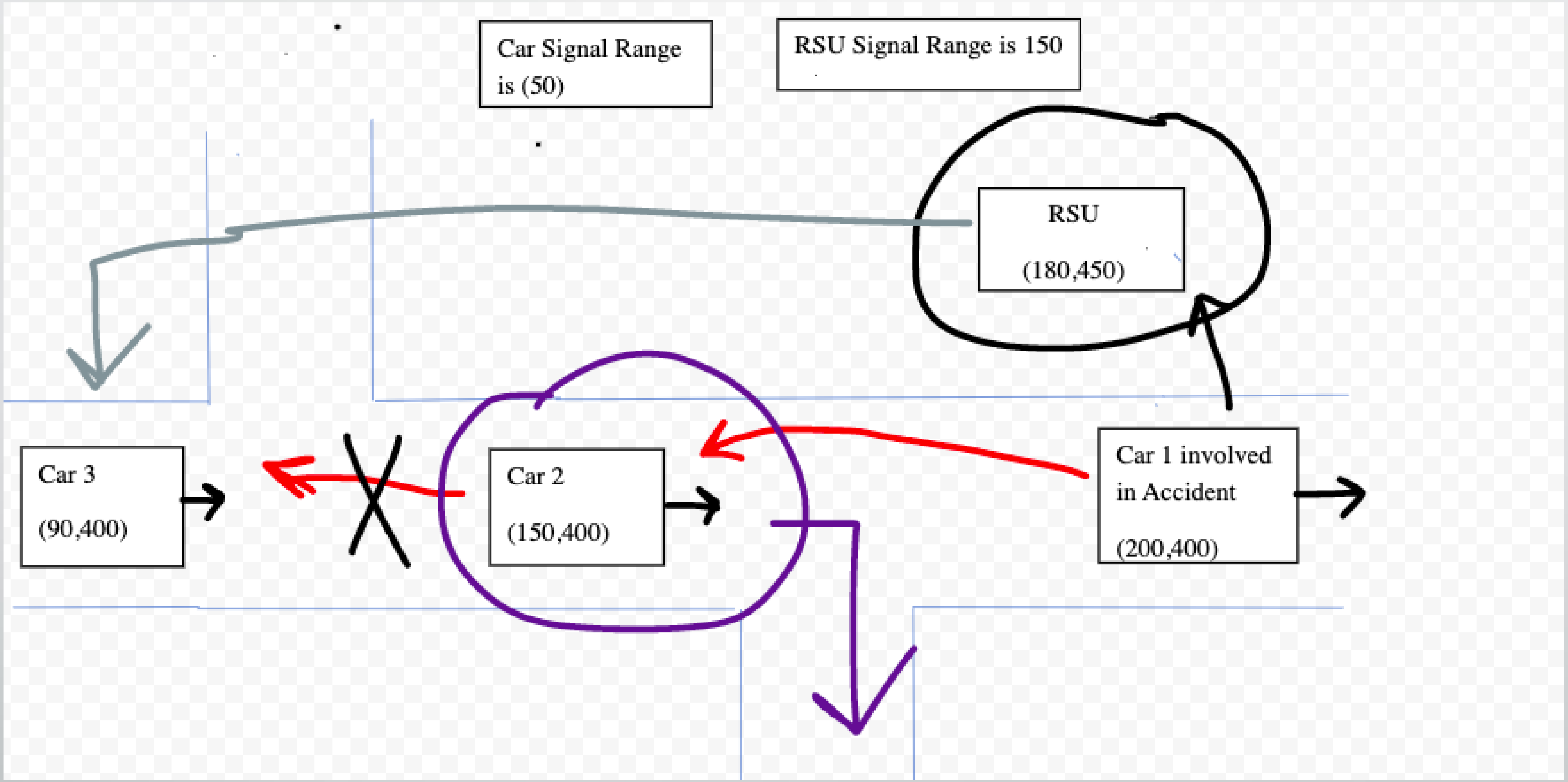
Code Implementation-11



```
if(reach<=150)
{
  if(visited[q]==true)
      System.out.println("Car "+q+" is in reach of RSU but it already gets V2V communication from

  else
  {
      System.out.println("Incident information passed from RSU to Car "+q);
      int xcord=arrayofcar[q].getXcoordinate();
      int ycord=arrayofcar[q].getYcoordinate();

  System.out.println("Car "+q+" moves to Left from coordinate ("+xcord+","+ycord+") to coordinate
  arrayofcar[q].setXcoordinate(xcord+25);
  arrayofcar[q].setYcoordinate(ycord+25);

  }
```

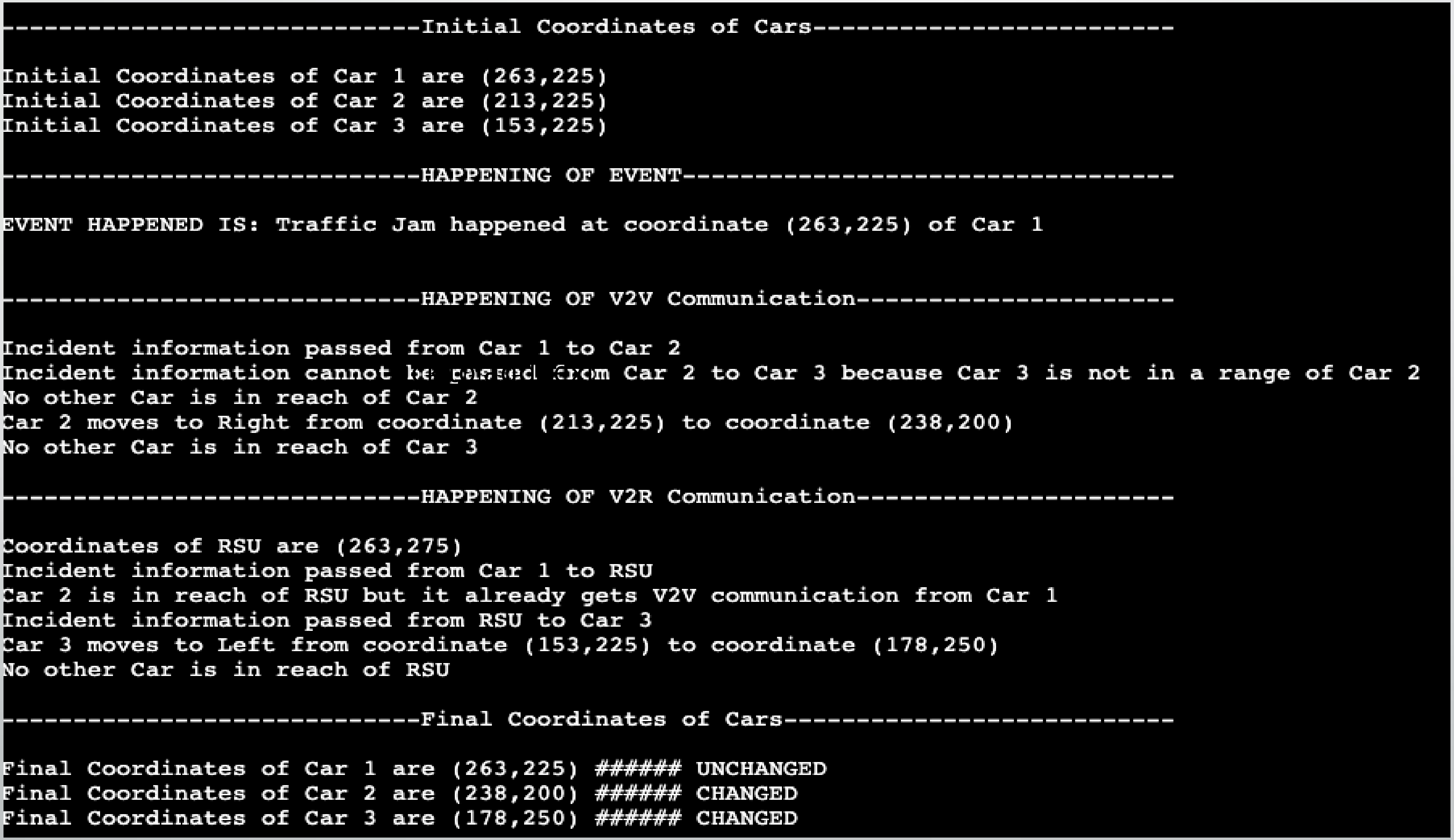
Sending V2R signal only if dis<=150 & that vehicle is not yet explored

Code Implementation-11



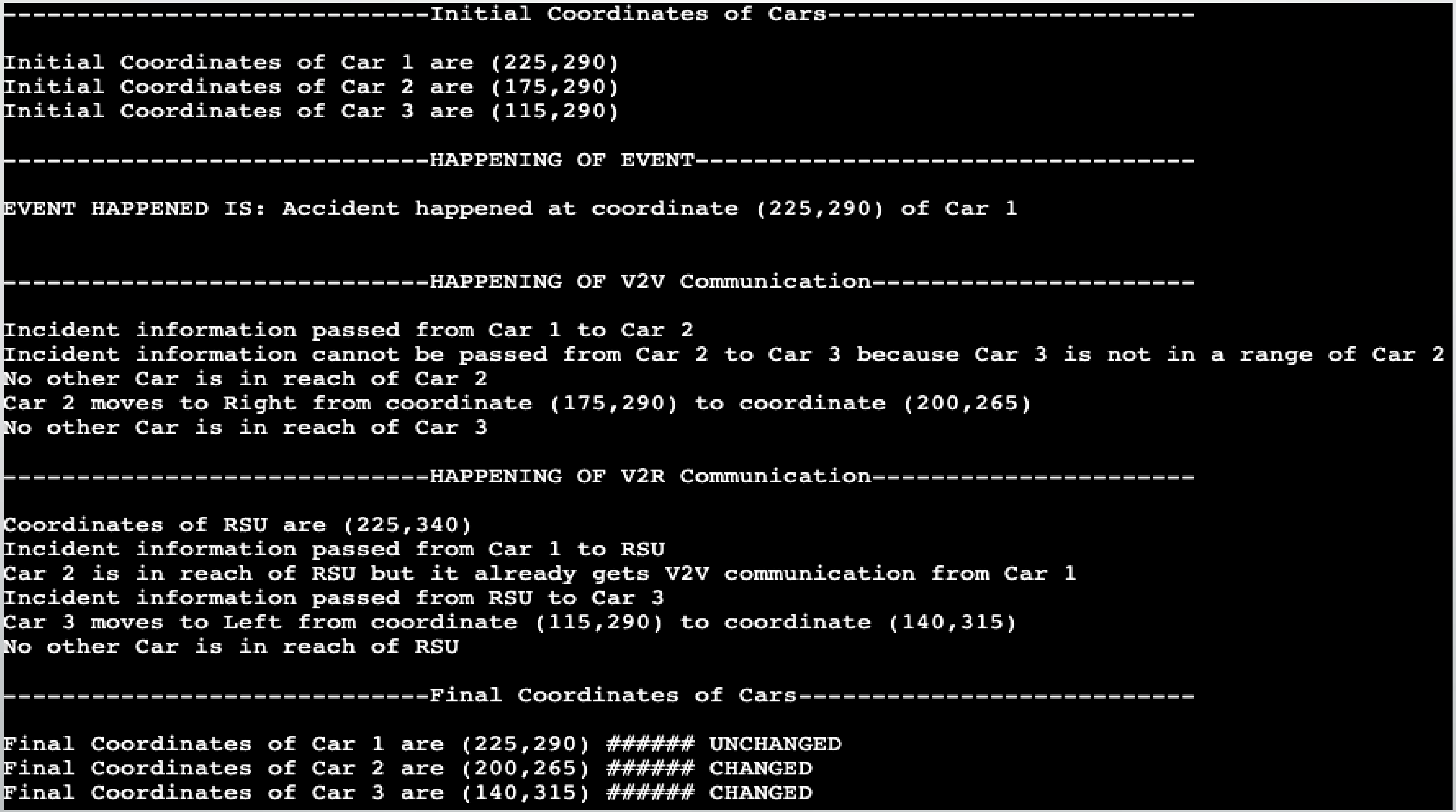
RSU send to Car 2 & Car 3, but Car 2 already explored. So, only R3 recieve

Output 1 (Info is True & Traffic Jam happened)



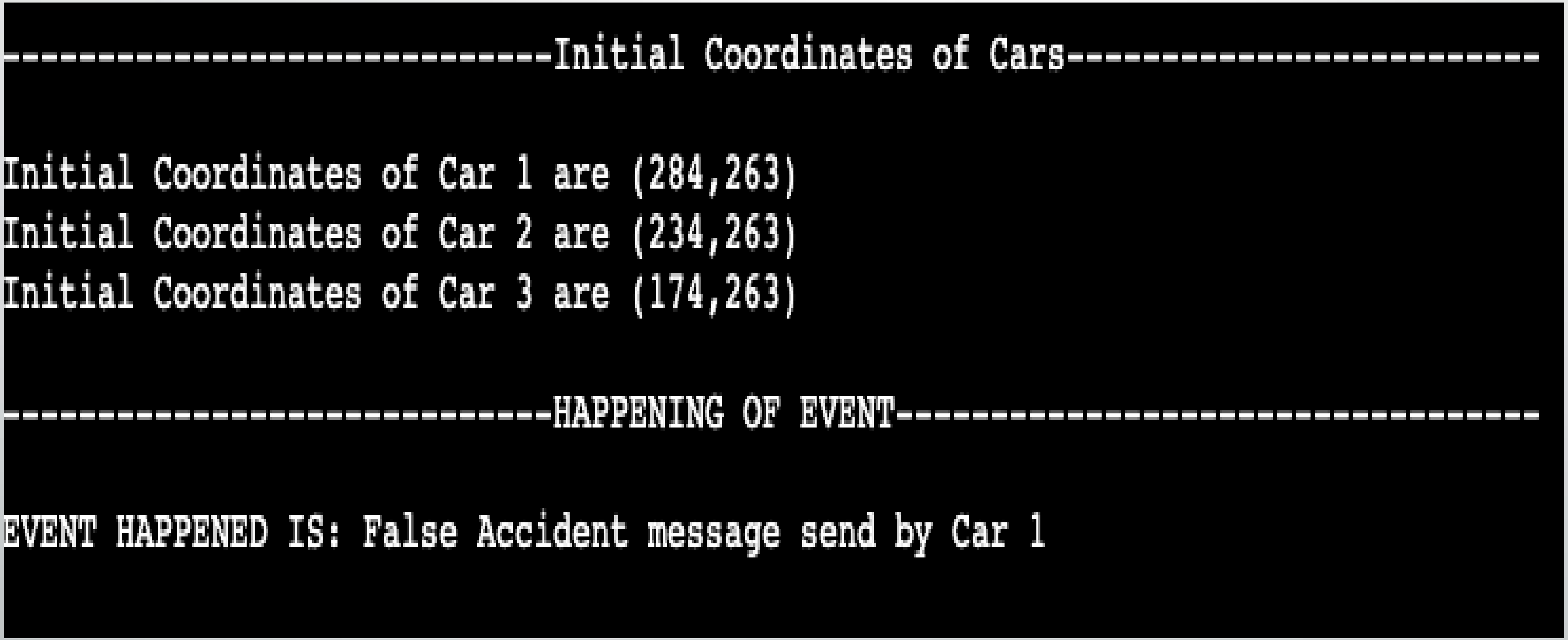
```
------------------------------Initial Coordinates of Cars-------------------------

Initial Coordinates of Car 1 are (263,225)
Initial Coordinates of Car 2 are (213,225)
Initial Coordinates of Car 3 are (153,225)

------------------------------HAPPENING OF EVENT----------------------------------

EVENT HAPPENED IS: Traffic Jam happened at coordinate (263,225) of Car 1


------------------------------HAPPENING OF V2V Communication----------------------

Incident information passed from Car 1 to Car 2
Incident information cannot be passed from Car 2 to Car 3 because Car 3 is not in a range of Car 2
No other Car is in reach of Car 2
Car 2 moves to Right from coordinate (213,225) to coordinate (238,200)
No other Car is in reach of Car 3

------------------------------HAPPENING OF V2R Communication----------------------

Coordinates of RSU are (263,275)
Incident information passed from Car 1 to RSU
Car 2 is in reach of RSU but it already gets V2V communication from Car 1
Incident information passed from RSU to Car 3
Car 3 moves to Left from coordinate (153,225) to coordinate (178,250)
No other Car is in reach of RSU

------------------------------Final Coordinates of Cars---------------------------

Final Coordinates of Car 1 are (263,225) ###### UNCHANGED
Final Coordinates of Car 2 are (238,200) ###### CHANGED
Final Coordinates of Car 3 are (178,250) ###### CHANGED
```

Output 2 (Info is True & Accident happened)



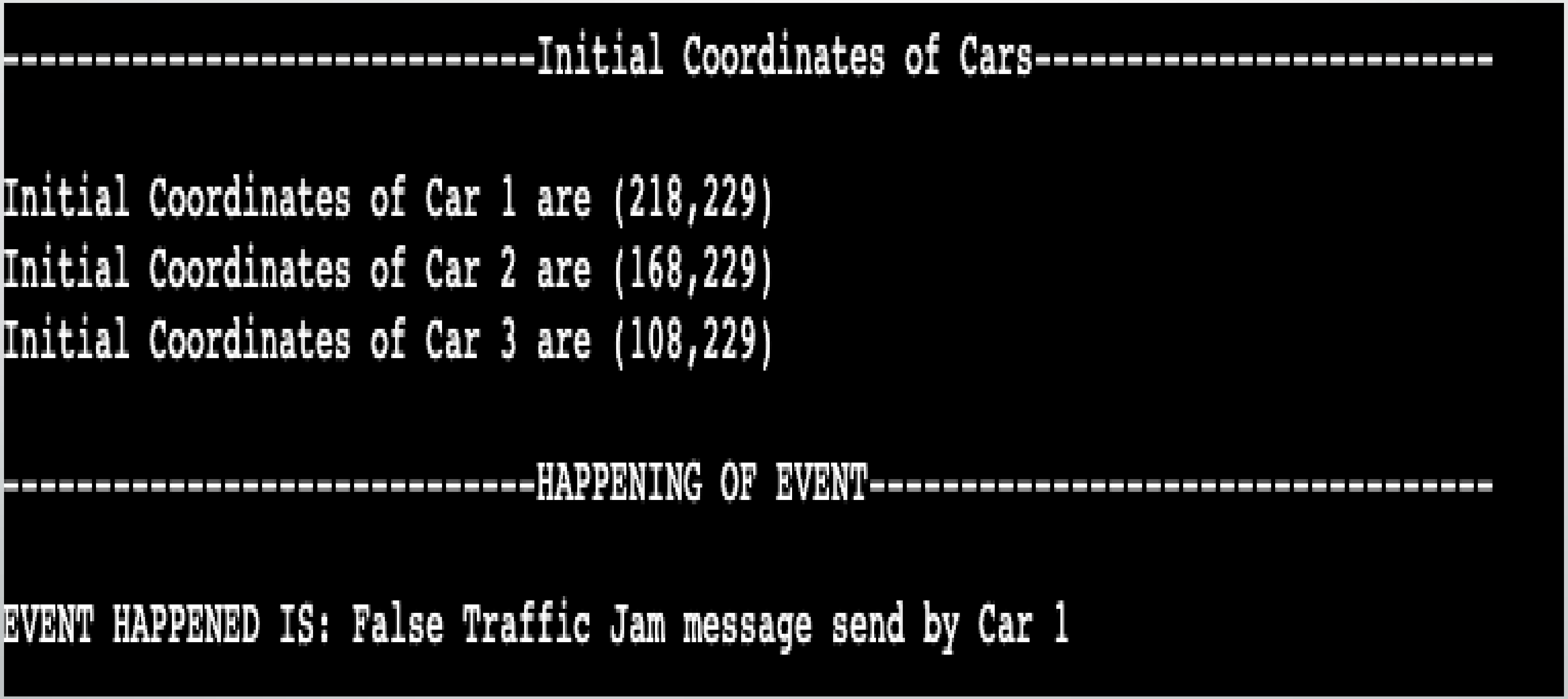
```
-----------------------------Initial Coordinates of Cars-------------------------

Initial Coordinates of Car 1 are (225,290)
Initial Coordinates of Car 2 are (175,290)
Initial Coordinates of Car 3 are (115,290)

-----------------------------HAPPENING OF EVENT----------------------------------

EVENT HAPPENED IS: Accident happened at coordinate (225,290) of Car 1


-----------------------------HAPPENING OF V2V Communication----------------------

Incident information passed from Car 1 to Car 2
Incident information cannot be passed from Car 2 to Car 3 because Car 3 is not in a range of Car 2
No other Car is in reach of Car 2
Car 2 moves to Right from coordinate (175,290) to coordinate (200,265)
No other Car is in reach of Car 3

-----------------------------HAPPENING OF V2R Communication----------------------

Coordinates of RSU are (225,340)
Incident information passed from Car 1 to RSU
Car 2 is in reach of RSU but it already gets V2V communication from Car 1
Incident information passed from RSU to Car 3
Car 3 moves to Left from coordinate (115,290) to coordinate (140,315)
No other Car is in reach of RSU

-----------------------------Final Coordinates of Cars---------------------------

Final Coordinates of Car 1 are (225,290) ###### UNCHANGED
Final Coordinates of Car 2 are (200,265) ###### CHANGED
Final Coordinates of Car 3 are (140,315) ###### CHANGED
```

Output 3 (Accident happened but Info is False)



```
------------------------------Initial Coordinates of Cars--------------------------


Initial Coordinates of Car 1 are (284,263)
Initial Coordinates of Car 2 are (234,263)
Initial Coordinates of Car 3 are (174,263)


------------------------------HAPPENING OF EVENT-----------------------------------


EVENT HAPPENED IS: False Accident message send by Car 1
```

Output 4 (Traffic Jam happened but Info is False)



```
-----------------------------Initial Coordinates of Cars-------------------------


Initial Coordinates of Car 1 are (218,229)
Initial Coordinates of Car 2 are (168,229)
Initial Coordinates of Car 3 are (108,229)


-----------------------------HAPPENING OF EVENT----------------------------------


EVENT HAPPENED IS: False Traffic Jam message send by Car 1
```

Code Links

Blockchain Implementation:

- https://github.com/rajat123456/Blockchain-Implementation-in-5G-Vehicular-Networks/blob/main/Blockchain_Implementation.java

Complete Code:

- https://github.com/rajat123456/Blockchain-Implementation-in-5G-Vehicular-Networks/blob/main/Final_Code.java

References

- https://medium.com/programmers-blockchain/create-simple-blockchain-java-tutorial-from-scratch-6eeed3cb03fa