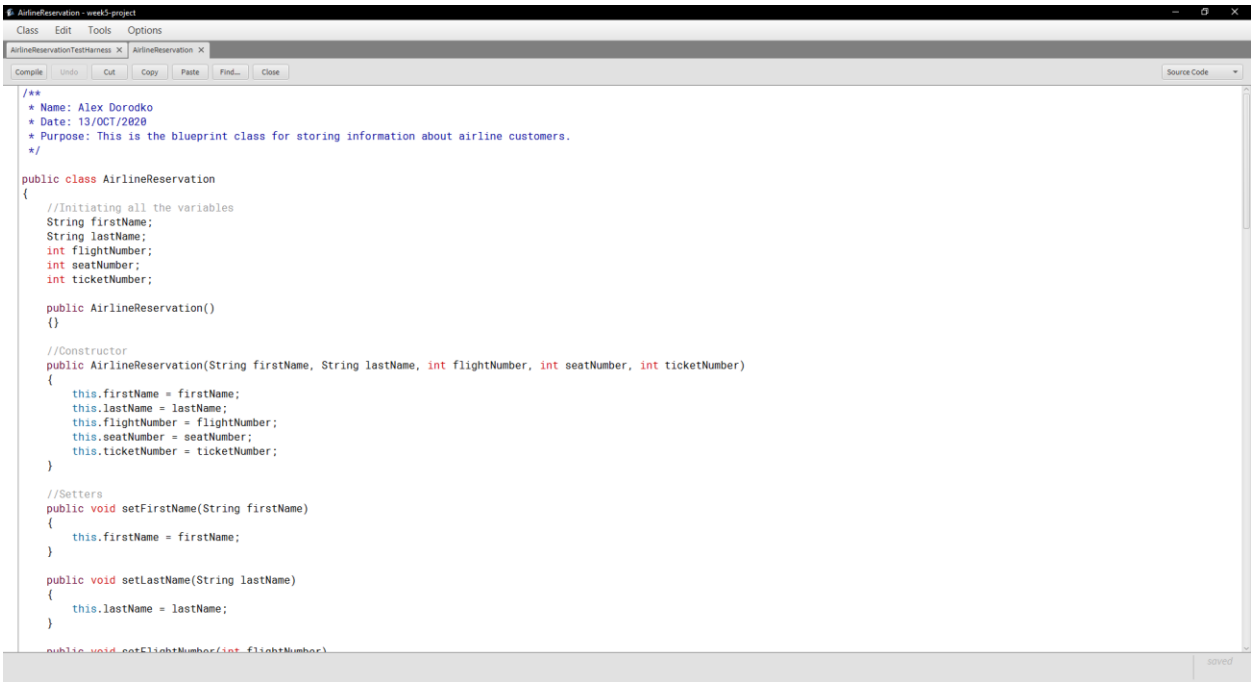
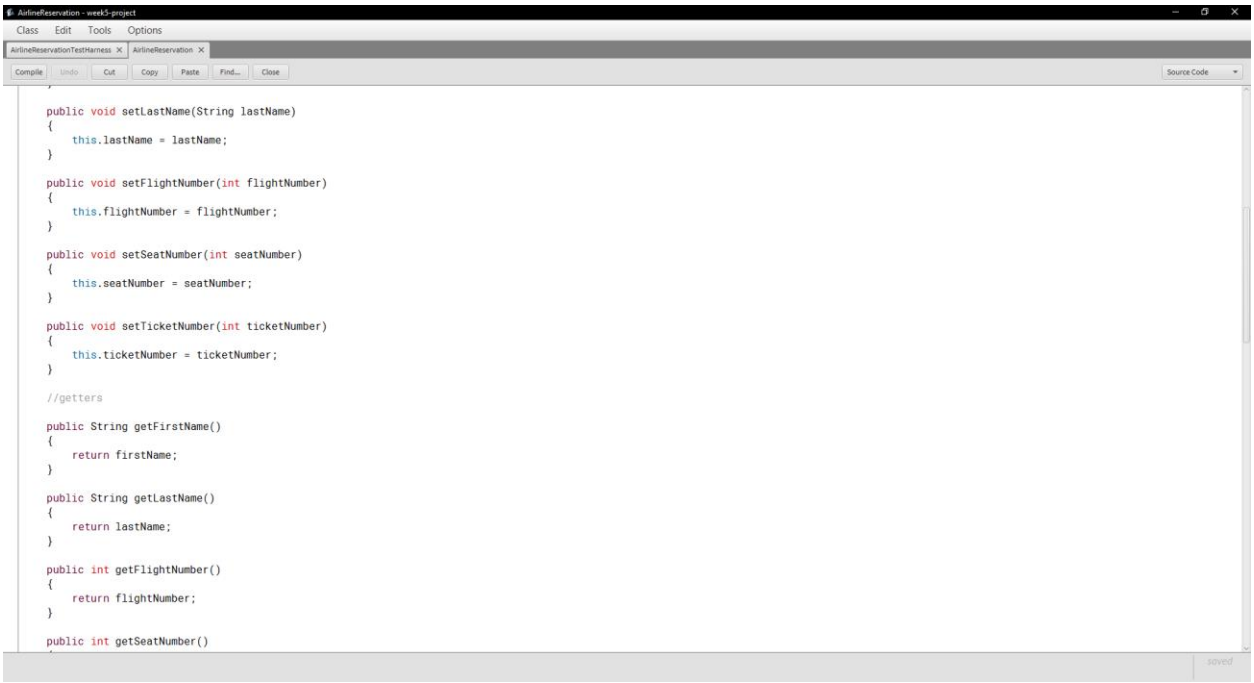
AirlineReservation.java

Part 1



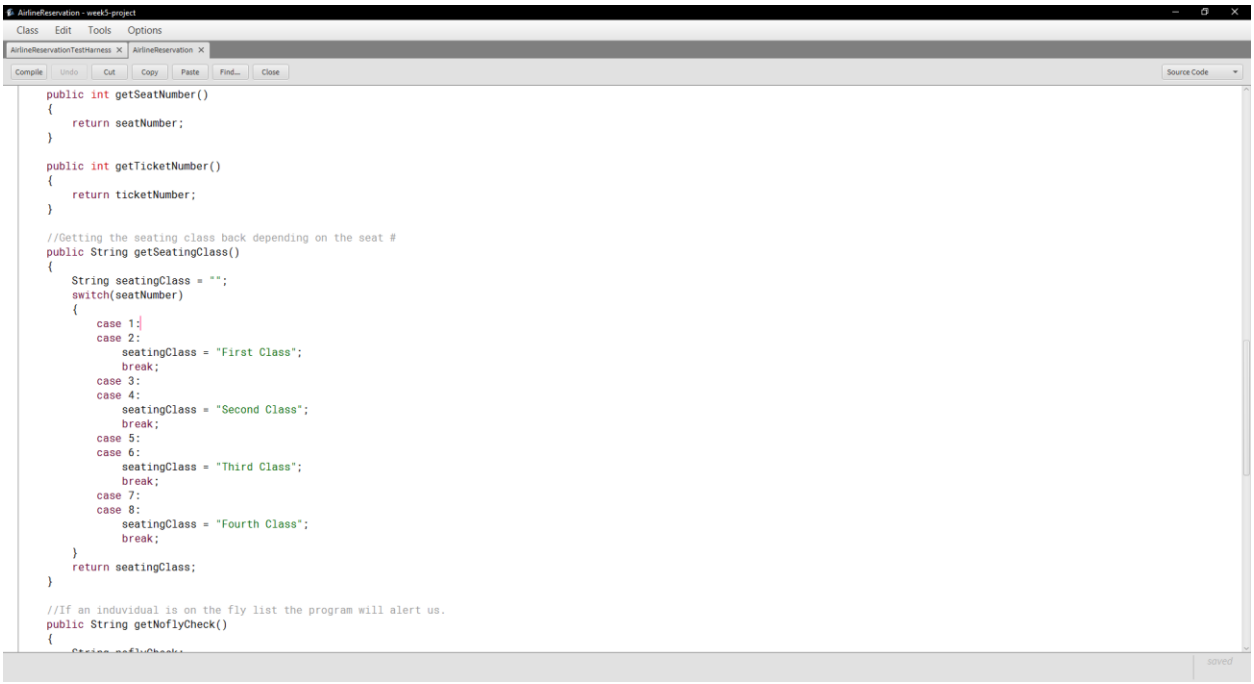

```java
/**
 * Name: Alex Dorodko
 * Date: 13/OCT/2020
 * Purpose: This is the blueprint class for storing information about airline customers.
 */

public class AirlineReservation
{
    //Initiating all the variables
    String firstName;
    String lastName;
    int flightNumber;
    int seatNumber;
    int ticketNumber;

    public AirlineReservation()
    {}

    //Constructor
    public AirlineReservation(String firstName, String lastName, int flightNumber, int seatNumber, int ticketNumber)
    {
        this.firstName = firstName;
        this.lastName = lastName;
        this.flightNumber = flightNumber;
        this.seatNumber = seatNumber;
        this.ticketNumber = ticketNumber;
    }

    //Setters
    public void setFirstName(String firstName)
    {
        this.firstName = firstName;
    }

    public void setLastName(String lastName)
    {
        this.lastName = lastName;
    }

    public void setFlightNumber(int flightNumber)
```

Part 2



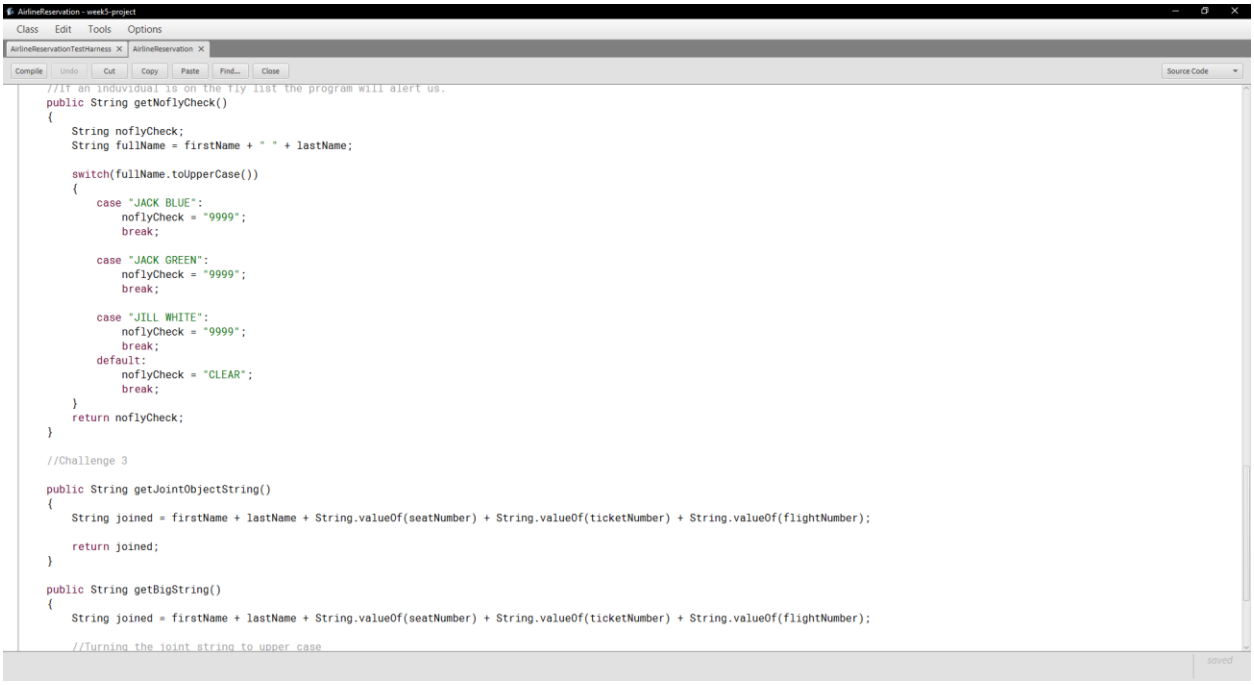

```java
    public void setLastName(String lastName)
    {
        this.lastName = lastName;
    }

    public void setFlightNumber(int flightNumber)
    {
        this.flightNumber = flightNumber;
    }

    public void setSeatNumber(int seatNumber)
    {
        this.seatNumber = seatNumber;
    }

    public void setTicketNumber(int ticketNumber)
    {
        this.ticketNumber = ticketNumber;
    }

    //getters

    public String getFirstName()
    {
        return firstName;
    }

    public String getLastName()
    {
        return lastName;
    }

    public int getFlightNumber()
    {
        return flightNumber;
    }

    public int getSeatNumber()
```

Part 3



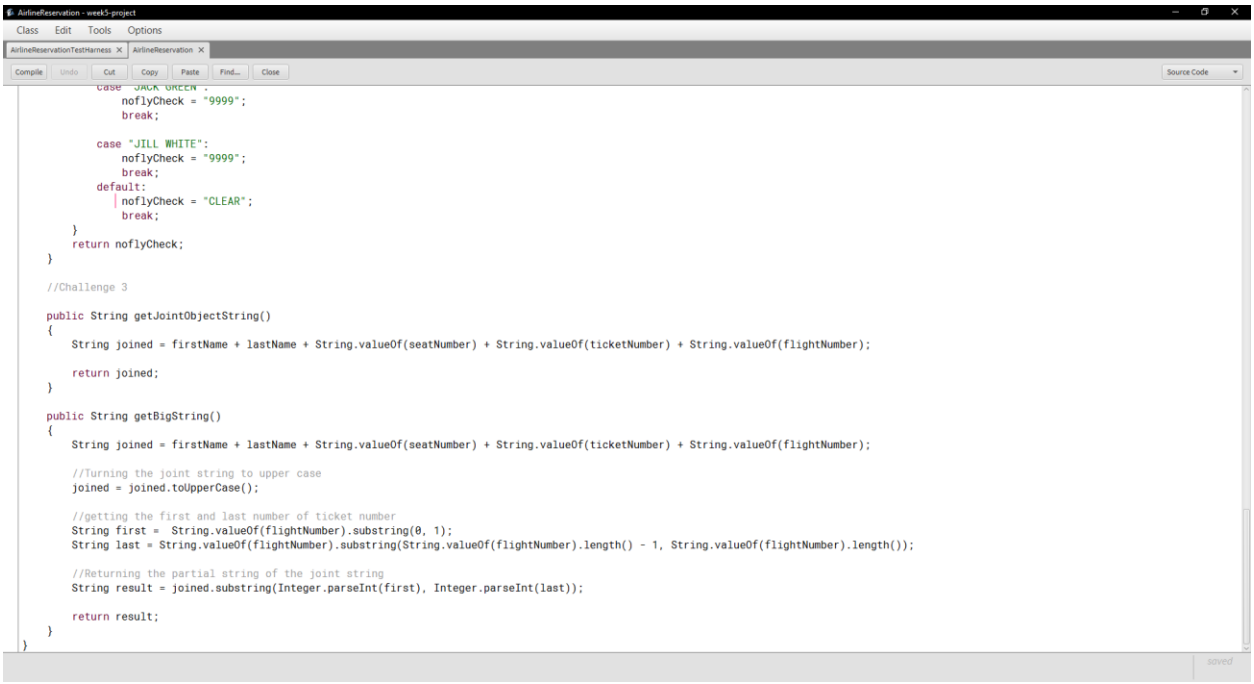

```
public int getSeatNumber()
{
    return seatNumber;
}

public int getTicketNumber()
{
    return ticketNumber;
}

//Getting the seating class back depending on the seat #
public String getSeatingClass()
{
    String seatingClass = "";
    switch(seatNumber)
    {
        case 1:
        case 2:
            seatingClass = "First Class";
            break;
        case 3:
        case 4:
            seatingClass = "Second Class";
            break;
        case 5:
        case 6:
            seatingClass = "Third Class";
            break;
        case 7:
        case 8:
            seatingClass = "Fourth Class";
            break;
    }
    return seatingClass;
}

//If an induvidual is on the fly list the program will alert us.
public String getNoflyCheck()
{
```

Part 4



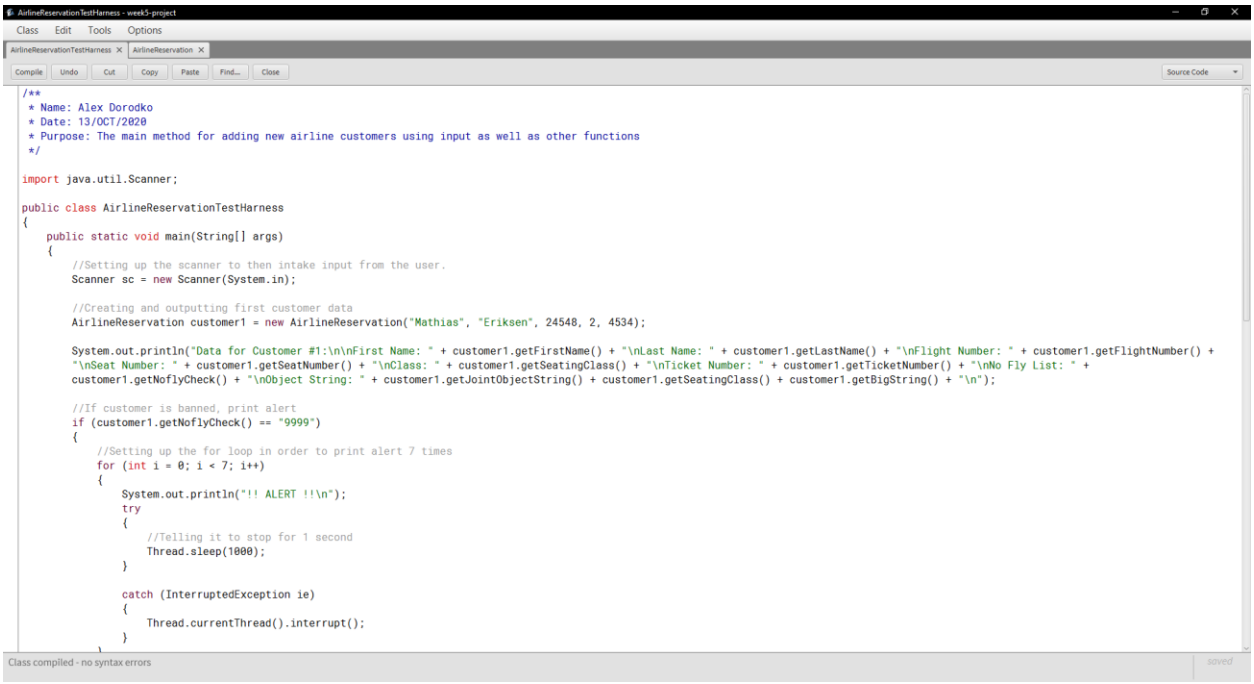

```
//If an induvidual is on the fly list the program will alert us.
public String getNoflyCheck()
{
    String noflyCheck;
    String fullName = firstName + " " + lastName;

    switch(fullName.toUpperCase())
    {
        case "JACK BLUE":
            noflyCheck = "9999";
            break;

        case "JACK GREEN":
            noflyCheck = "9999";
            break;

        case "JILL WHITE":
            noflyCheck = "9999";
            break;
        default:
            noflyCheck = "CLEAR";
            break;
    }
    return noflyCheck;
}

//Challenge 3

public String getJointObjectString()
{
    String joined = firstName + lastName + String.valueOf(seatNumber) + String.valueOf(ticketNumber) + String.valueOf(flightNumber);

    return joined;
}

public String getBigString()
{
    String joined = firstName + lastName + String.valueOf(seatNumber) + String.valueOf(ticketNumber) + String.valueOf(flightNumber);

    //Turning the joint string to upper case
```

Part 5


AirlineReservation - week3-project
Class Edit Tools Options
AirlineReservationTestHarness
AirlineReservation
Compile Undo Cut Copy Paste Find... Close
Source Code
case "JACK GREEN":
noflyCheck = "9999";
break;
case "JILL WHITE":
noflyCheck = "9999";
break;
default:
noflyCheck = "CLEAR";
break;
}
return noflyCheck;
}
//Challenge 3
public String getJointObjectString()
{
String joined = firstName + lastName + String.valueOf(seatNumber) + String.valueOf(ticketNumber) + String.valueOf(flightNumber);
return joined;
}
public String getBigString()
{
String joined = firstName + lastName + String.valueOf(seatNumber) + String.valueOf(ticketNumber) + String.valueOf(flightNumber);
//Turning the joint string to upper case
joined = joined.toUpperCase();
//getting the first and last number of ticket number
String first = String.valueOf(flightNumber).substring(0, 1);
String last = String.valueOf(flightNumber).substring(String.valueOf(flightNumber).length() - 1, String.valueOf(flightNumber).length());
//Returning the partial string of the joint string
String result = joined.substring(Integer.parseInt(first), Integer.parseInt(last));
return result;
}
}
saved

AirlineReservationTestHarness.java

Part 1

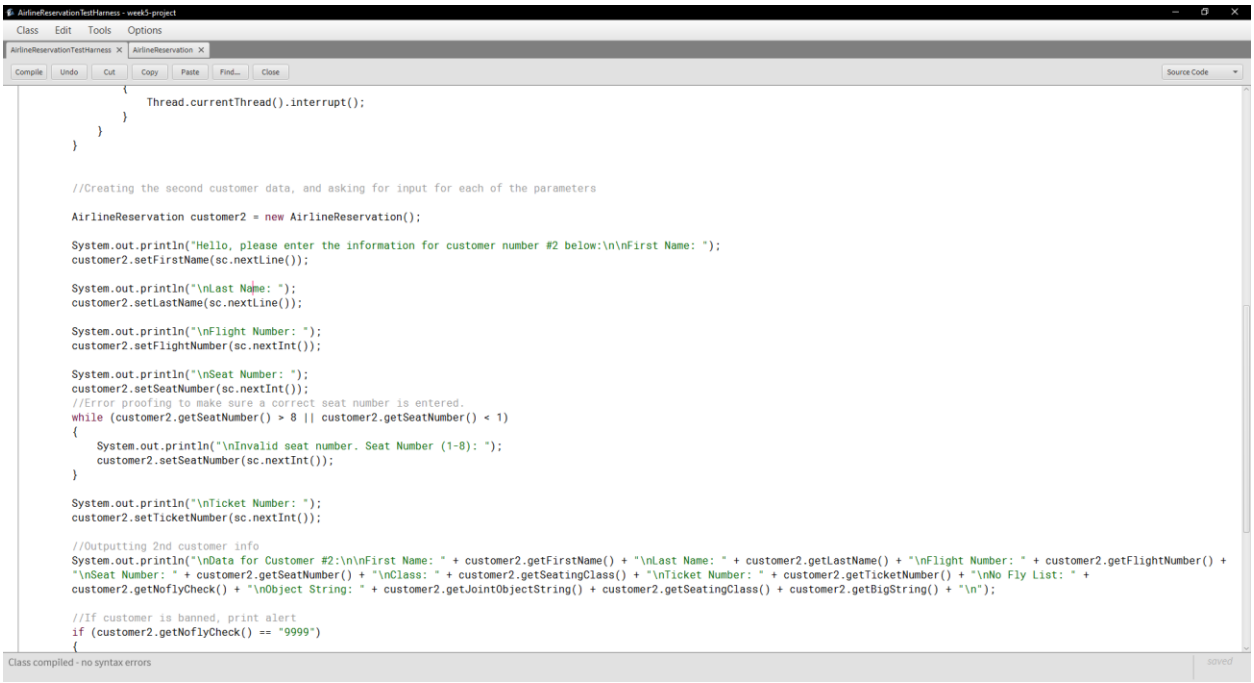


```
/**
 * Name: Alex Dorodko
 * Date: 13/OCT/2020
 * Purpose: The main method for adding new airline customers using input as well as other functions
 */

import java.util.Scanner;

public class AirlineReservationTestHarness
{
    public static void main(String[] args)
    {
        //Setting up the scanner to then intake input from the user.
        Scanner sc = new Scanner(System.in);

        //Creating and outputting first customer data
        AirlineReservation customer1 = new AirlineReservation("Mathias", "Eriksen", 24548, 2, 4534);

        System.out.println("Data for Customer #1:\n\nFirst Name: " + customer1.getFirstName() + "\nLast Name: " + customer1.getLastName() + "\nFlight Number: " + customer1.getFlightNumber() +
        "\nSeat Number: " + customer1.getSeatNumber() + "\nClass: " + customer1.getSeatingClass() + "\nTicket Number: " + customer1.getTicketNumber() + "\nNo Fly List: " +
        customer1.getNoflyCheck() + "\nObject String: " + customer1.getJointObjectString() + customer1.getSeatingClass() + customer1.getBigString() + "\n");

        //If customer is banned, print alert
        if (customer1.getNoflyCheck() == "9999")
        {
            //Setting up the for loop in order to print alert 7 times
            for (int i = 0; i < 7; i++)
            {
                System.out.println("!! ALERT !!\n");
                try
                {
                    //Telling it to stop for 1 second
                    Thread.sleep(1000);
                }

                catch (InterruptedException ie)
                {
                    Thread.currentThread().interrupt();
                }
```

Class compiled - no syntax errors

saved

Part 2

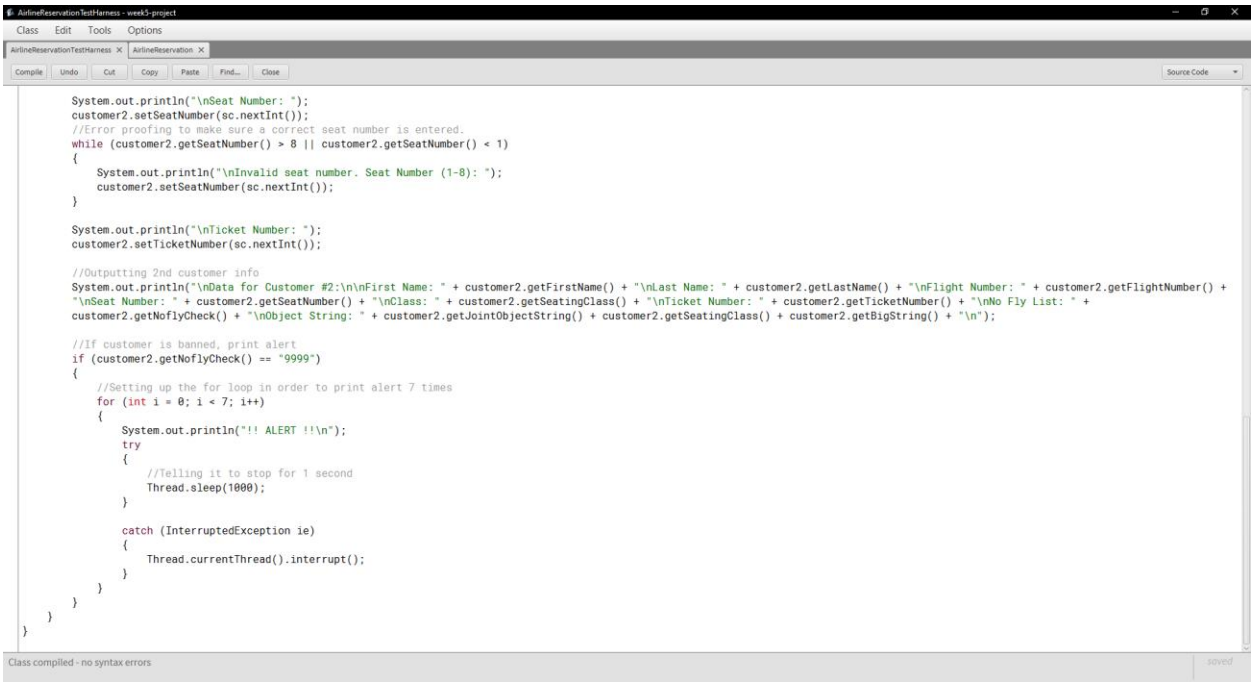


```
                {
                    Thread.currentThread().interrupt();
                }
            }
        }


        //Creating the second customer data, and asking for input for each of the parameters

        AirlineReservation customer2 = new AirlineReservation();

        System.out.println("Hello, please enter the information for customer number #2 below:\n\nFirst Name: ");
        customer2.setFirstName(sc.nextLine());

        System.out.println("\nLast Name: ");
        customer2.setLastName(sc.nextLine());

        System.out.println("\nFlight Number: ");
        customer2.setFlightNumber(sc.nextInt());

        System.out.println("\nSeat Number: ");
        customer2.setSeatNumber(sc.nextInt());
        //Error proofing to make sure a correct seat number is entered.
        while (customer2.getSeatNumber() > 8 || customer2.getSeatNumber() < 1)
        {
            System.out.println("\nInvalid seat number. Seat Number (1-8): ");
            customer2.setSeatNumber(sc.nextInt());
        }

        System.out.println("\nTicket Number: ");
        customer2.setTicketNumber(sc.nextInt());

        //Outputting 2nd customer info
        System.out.println("\nData for Customer #2:\n\nFirst Name: " + customer2.getFirstName() + "\nLast Name: " + customer2.getLastName() + "\nFlight Number: " + customer2.getFlightNumber() +
        "\nSeat Number: " + customer2.getSeatNumber() + "\nClass: " + customer2.getSeatingClass() + "\nTicket Number: " + customer2.getTicketNumber() + "\nNo Fly List: " +
        customer2.getNoflyCheck() + "\nObject String: " + customer2.getJointObjectString() + customer2.getSeatingClass() + customer2.getBigString() + "\n");

        //If customer is banned, print alert
        if (customer2.getNoflyCheck() == "9999")
        {
```

Class compiled - no syntax errors

saved

Part 3


AirlineReservationTestHarness - week3-project
Class Edit Tools Options
AirlineReservationTestHarness
AirlineReservation
Compile Undo Cut Copy Paste Find... Close
Source Code
System.out.println("\nSeat Number: ");
customer2.setSeatNumber(sc.nextInt());
//Error proofing to make sure a correct seat number is entered.
while (customer2.getSeatNumber() > 8 || customer2.getSeatNumber() < 1)
{
System.out.println("\nInvalid seat number. Seat Number (1-8): ");
customer2.setSeatNumber(sc.nextInt());
}
System.out.println("\nTicket Number: ");
customer2.setTicketNumber(sc.nextInt());
//Outputting 2nd customer info
System.out.println("\nData for Customer #2:\n\nFirst Name: " + customer2.getFirstName() + "\nLast Name: " + customer2.getLastName() + "\nFlight Number: " + customer2.getFlightNumber() +
"\nSeat Number: " + customer2.getSeatNumber() + "\nClass: " + customer2.getSeatingClass() + "\nTicket Number: " + customer2.getTicketNumber() + "\nNo Fly List: " +
customer2.getNoflyCheck() + "\nObject String: " + customer2.getJointObjectString() + customer2.getSeatingClass() + customer2.getBigString() + "\n");
//If customer is banned, print alert
if (customer2.getNoflyCheck() == "9999")
{
//Setting up the for loop in order to print alert 7 times
for (int i = 0; i < 7; i++)
{
System.out.println("!! ALERT !!\n");
try
{
//Telling it to stop for 1 second
Thread.sleep(1000);
}
catch (InterruptedException ie)
{
Thread.currentThread().interrupt();
}
}
}
}
}
Class compiled - no syntax errors
saved