



NAME: Junaid khan

ID#: 16027

MODULE: Bachelors {Software Engineering}

SEMESTER: 2nd

SECTION: A

SUBJECT: Object Oriented Programming

INSTRUCTOR: M.Ayub Khan

Q1. What is Class and role of object in a Class, explain in detail with the help of a suitable program?

ANSWER:

Classes and Objects in Java

Classes and Objects are basic concepts of Object-Oriented Programming which revolve around the real-life entities.

Class:

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

1. **Modifiers** : A class can be public or has default access (Refer [this](#) for details).
2. **Class name**: The name should begin with a initial letter (capitalized by convention).
3. **Superclass (if any)**: The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.
4. **Interfaces (if any)**: A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. **Body**: The class body surrounded by braces, {}.

Constructors are used for initializing new objects. Fields are variables that provides the state of the class and its objects, and methods are used to implement the behavior of the class and its objects.

There are various types of classes that are used in real time applications such as **nested classes**, **anonymous classes**, **lambda expressions**.

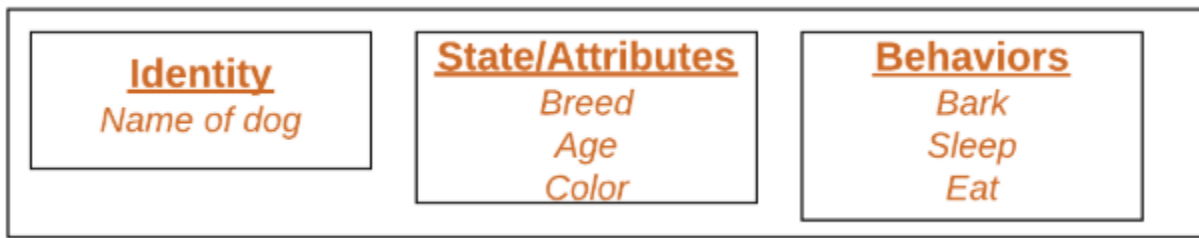
Object:

The **Object class** defines the basic state and behavior that all **objects** must have, such as the ability to compare oneself to another **object**, to convert to a string, to wait on a condition variable, to notify other **objects** that a condition variable has changed, and to return the **object's class**.

It is a basic unit of Object-Oriented Programming and represents the real-life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of:

1. **State**: It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behavior**: It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity**: It gives a unique name to an object and enables one object to interact with other objects

Example of an object: dog:

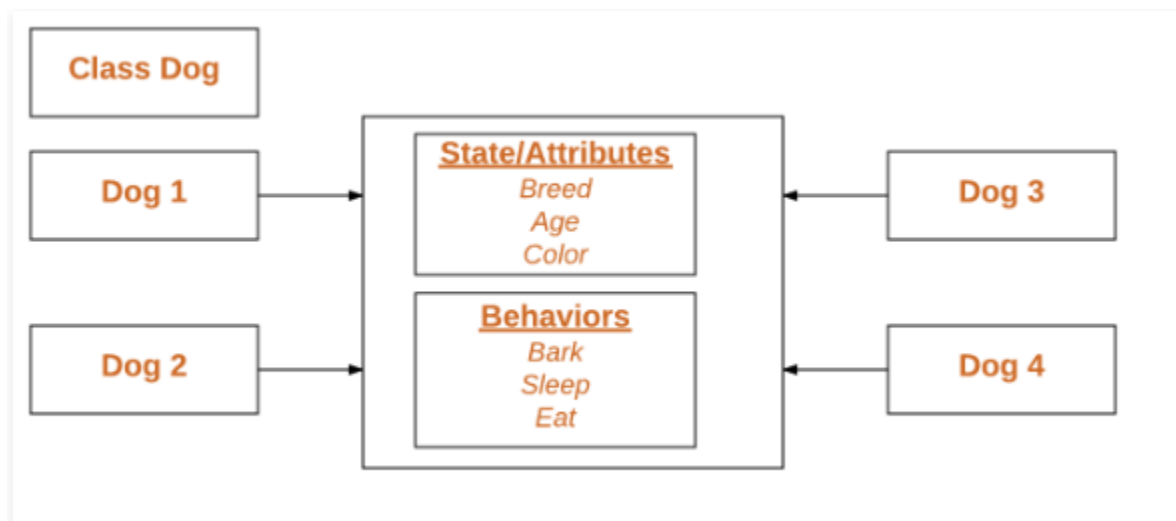


Objects correspond to things found in the real world. For example, a graphics program may have objects such as "circle", "square", "menu". An online shopping system might have objects such as "shopping cart", "customer", and "product".

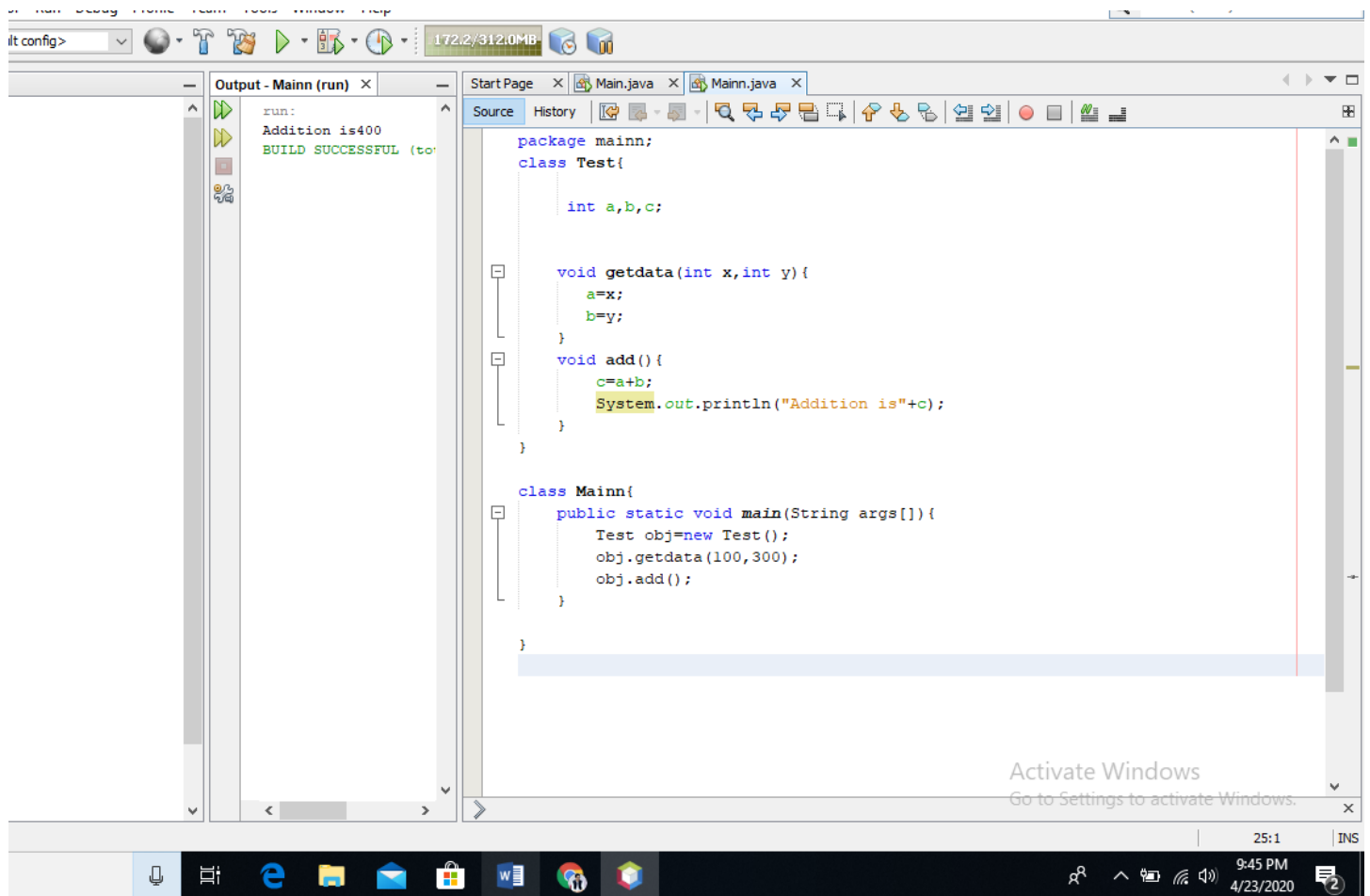
Declaring Objects (Also called instantiating a class)

When an object of a class is created, the class is said to be **instantiated**. All the instances share the attributes and the behavior of the class. But the values of those attributes, i.e. the state are unique for each object. A single class may have any number of instances.

Example:



RUN PROGRAM IN NETBEANS AND ITS OUTPUT:



Q2. Write a program about table printing which takes input from the user on the basis of OOP and explain in detail.

ANSWER:

Program for Table:

```
1. package main;
2. import java.util.Scanner;
3. class Table{
4.     int n, j,i;
5.     Scanner cin=new Scanner(System.in);
6.     void getdata(){
7.         System.out.println("enter a number of table");
8.         n=cin.nextInt();
9.         System.out.println("enter a length of table");
10.        j=cin.nextInt();
11.    }
12.    void disp(){
13.        for(i=0 ; i<=j ; i++){
14.            System.out.println(n+"*" +i+"="+n*i);
15.        }
16.    }
17. }

18. class Main{
19.     public static void main(String args[]){
20.         Table obj=new Table();
21.         obj.getdata();
22.         obj.disp();
23.     }
24. }
```

EXPLANATION:

1. in the first line we create the package with the "Main" name.
2. in the second line, we called the Scanner class through "import key" because the scanner class Some Methods that is used to take input from the user.
3. in the 3rd line we create the class with the "Table" name and also start the class with "{" bracket.
4. in the 4 Line we declared three fields or variables with the integer data type.
5. in the 5 Line we create the scanner object.
6. in the 6 line we create the methods or function with the "getdata ()" name and also start the methods With "{"bracket.
7. in the 7 line we display a message to user enter a number for the table.

8. in the 8 line we also take the value for the table from the user.
9. in the 9 Line we display a message to user "enter a number for the length of table".
10. in the line 10 we also take the value for the length of table from the user.
11. in the line 11 we closed the Methods with "}" bracket.
12. in the 12 line we create the Method with the "disp()" name and also start the methods with "{" bracket.
13. in the 13 Line we used "for – Loop".
14. in the 14 Line we print fields values and print some symbols e.g. *, =.
15. in the 15, 16, 17 Lines we closed the "for Loop", "disp Method" and "Table class" with separate "}" brackets.
16. In the 18 Line we create the class with the "Main" name and also start the class with "{" bracket.
17. in the 19 line we create the "main function" and also start the method with "{" bracket.
18. in the 20 line we create object for Table class to call the table class methods.
19. in the 21 line we called the "getdata()" method in the main function.
20. in the 22 line we called the "disp()" method in the main function.
21. in the 23 and 24 lines we closed the "main method" and "Main class" with separate "}" brackets.

Output:

enter a number of table

2

enter a length of table

10

2*0=0

2*1=2

2*2=4

2*3=6

2*4=8

2*5=10

2*6=12

2*7=14

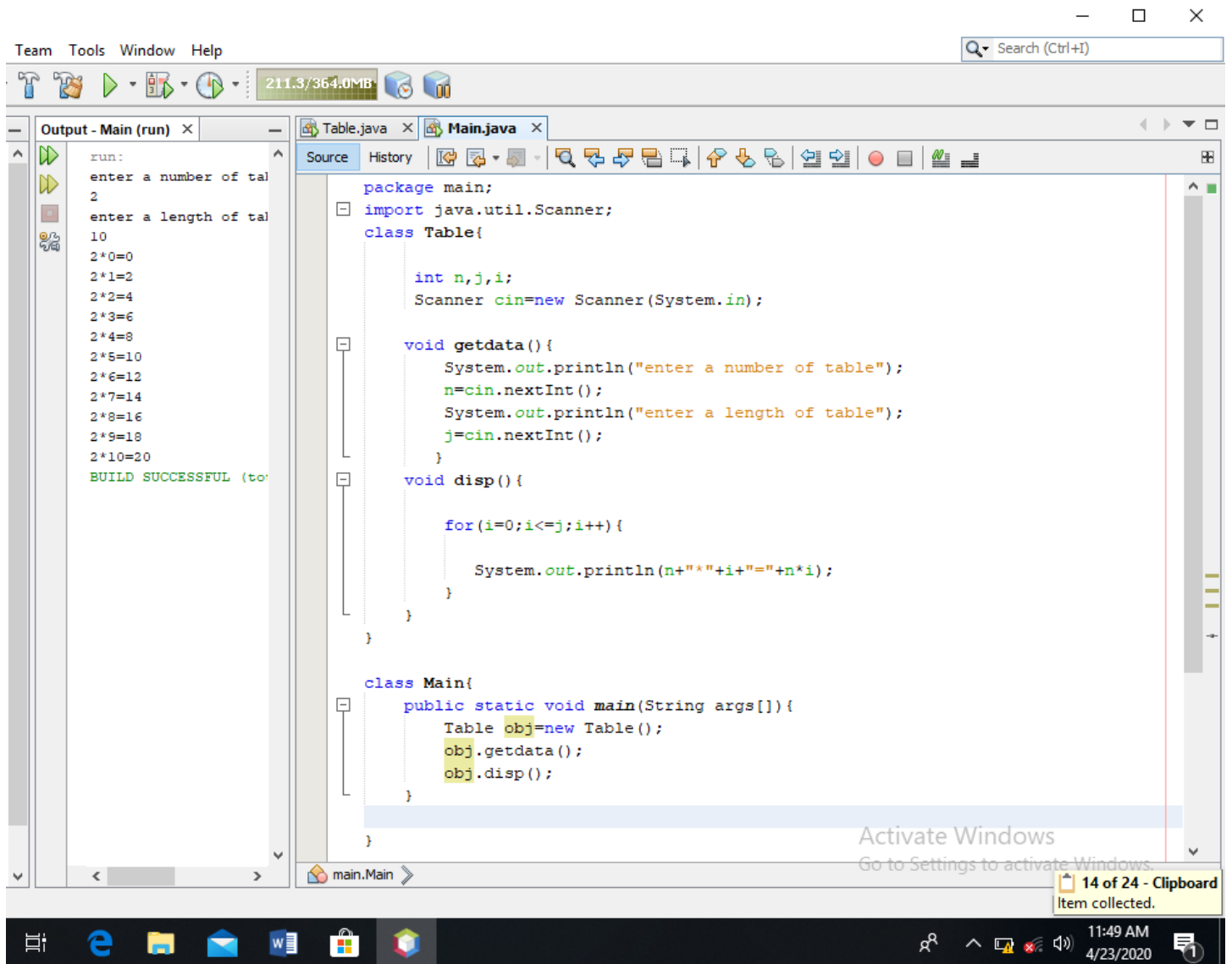
2*8=16

2*9=18

2*10=20

BUILD SUCCESSFUL (to

RUN PROGRAM IN NETBEANS AND ITS OUTPUT:



Q3. Write a program about any 2 cars which can calculate the performance of both of them and explain in detail.

ANSWER:

program about 2 cars which can calculate the performance of both of them:

```
package car;
```

```
import java.util.Scanner;
```

```
public class Car {
```

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

```
        Car1 pia= new Car1();
```

```
        Car2 obj= new Car2();
```

```
        Scanner py= new Scanner(System.in);
```

```
        System.out.println("Enter the no of Rides for the car 1 ");
```

```
        pia.ride1=py.nextInt();
```

```
        System.out.println("Enter the no of Rides for the car 2 ");
```

```
        obj.ride2=py.nextInt();
```

```
        System.out.println("Enter the Fuel Capacity of the car 1");
```

```
        pia.fca1=py.nextDouble();
```

```
        System.out.println("Enter the Fuel Capacity of the car 2");
```

```
        obj.fca2=py.nextDouble();
```

```
        System.out.println("Enter the Fuel Consumption of the car 1");
```

```
        pia.fc1=py.nextDouble();
```

```
        System.out.println("Enter the Fuel Consumption of the car 2");
```

```
        obj.fc2=py.nextDouble();
```

```
        System.out.println("");
```

```
        System.out.println("No of Rides in the car 1 "+>"+pia.ride1);
```

```
        System.out.println("Total Fuel Capacity of the car 1 "+>"+pia.fca1);
```

```
        System.out.println("Fuel Consumption of the car 1 "+>"+pia.fc1);
```

```
        System.out.println("Performance 1st method of the car 1 " );
```

```
        pia.perfom1();
```

```
        System.out.println("Performance 2nd method of the car 1 "+>"+ pia.perfo1() );
```

```
        System.out.println("Driving time of the car 1 "+>"+ pia.ct1(3.5) );
```

```
        System.out.println("");
```

```
        System.out.println("No of Rides in the car 2 "+>"+obj.ride2);
```

```
        System.out.println("Total Fuel Capacity of the car 2 "+>"+obj.fca2);
```

```
        System.out.println("Fuel Consumption of the car 2 "+>"+obj.fc2);
```

```
        System.out.println("Performance 1st method of the car2 " );
```

```
        obj.perfom();
```

```
        System.out.println("Performance 2nd method of the car 2 "+>"+ pia.perfo1() );
```

```
        System.out.println("Driving time of the car 2 "+>"+ pia.ct1(3.5) );
```

```
    }
```

```
}
```

```
class Car1
```

```
{
```



```

        int ride1;
        double fca1;
        double fc1;
void perform1()
{
        double p;
        p= fca1/fc1;
        System.out.println("Performance "+p);
}
double perfo1()
{
        double p;
        p= fca1/fc1;
        return p;
}
double ct1(double t)
{
        return fc1*t ;
}
}
class Car2
{
        int ride2;
        double fca2;
        double fc2;
void perform()
{
        double p;
        p= fca2/fc2;
        System.out.println("Performance "+p);
}
double perfo()
{
        double p;
        p= fca2/fc2;
        return p;
}
double ft(double t)
{
        return fc2*t ;
}
}

```

Explanation:

In this program we created three classes "class car1", "class car2" and "class car", In the (class car1) we declared three fields (ride1, fca1,fc1) and three methods(perform1(), perfo1() , ct1(double t)) for calculating the car1 performance, and In the (class car2) we also declared three fields (ride2, fca2,fc2) and three

methods(perfom(), perfo() , ct(double t)) for calculating the car2 performance, in the (class car) first we create the scanner object and then we create object for “car1 class” and “car2 class” to call the car1 class methods and car2 class Methods. After that we display a message to user to take input from the user.

Output:

run:

Enter the no of Rides for the car 1

5

Enter the no of Rides for the car 2

6

Enter the Fuel Capacity of the car 1

20

Enter the Fuel Capacity of the car 2

26

Enter the Fuel Consumption of the car 1

12

Enter the Fuel Consumption of the car 2

14

No of Rides in the car 1 =>5

Total Fuel Capacity of the car 1 =>20.0

Fuel Consumption of the car 1 =>12.0

Performance 1st method of the car 1

Performance 1.6666666666666667

Performance 2nd method of the car 1 =>1.6666666666666667

Driving time of the car 1 =>42.0

No of Rides in the car 2 =>6

Total Fuel Capacity of the car 2 =>26.0

Fuel Consumption of the car 2 =>14.0

Performance 1st method of the car2

Performance 1.8571428571428572

Performance 2nd method of the car 2 =>1.666666666666667

Driving time of the car 2 =>42.0

BUILD SUCCESSFUL (total time: 41 seconds)

RUN PROGRAM IN NETBEANS AND ITS OUTPUT:

```
Output - Car (run) x
run:
Enter the no of Rides for the car 1
5
Enter the no of Rides for the car 2
6
Enter the Fuel Capacity of the car 1
20
Enter the Fuel Capacity of the car 2
26
Enter the Fuel Consumption of the car 1
12
Enter the Fuel Consumption of the car 2
14

No of Rides in the car 1 =>5
Total Fuel Capacity of the car 1 =>20.0
Fuel Consumption of the car 1 =>12.0
Performance 1st method of the car 1
Performance 1.666666666666667
Performance 2nd method of the car 1 =>1.66
Driving time of the car 1 =>42.0

No of Rides in the car 2 =>6
Total Fuel Capacity of the car 2 =>26.0
Fuel Consumption of the car 2 =>14.0
Performance 1st method of the car2
Performance 1.8571428571428572
Performance 2nd method of the car 2 =>1.66
Driving time of the car 2 =>42.0
BUILD SUCCESSFUL (total time: 16 seconds)

package car;

import java.util.Scanner;
public class Car {
    public static void main(String[] args) {
        Car1 pia= new Car1();
        Car2 obj= new Car2();
        Scanner py= new Scanner(System.in);
        System.out.println("Enter the no of Rides for the car 1 ");
        pia.ride1=py.nextInt();
        System.out.println("Enter the no of Rides for the car 2 ");
        obj.ride2=py.nextInt();
        System.out.println("Enter the Fuel Capacity of the car 1");
        pia.fca1=py.nextDouble();
        System.out.println("Enter the Fuel Capacity of the car 2");
        obj.fca2=py.nextDouble();
        System.out.println("Enter the Fuel Consumption of the car 1");
        pia.fc1=py.nextDouble();
        System.out.println("Enter the Fuel Consumption of the car 2");
        obj.fc2=py.nextDouble();

        System.out.println("");

        System.out.println("No of Rides in the car 1 "+=>+pia.ride1);
        System.out.println("Total Fuel Capacity of the car 1 "+=>+pia.fca1);
        System.out.println("Fuel Consumption of the car 1 "+=>+pia.fc1);
        System.out.println("Performance 1st method of the car 1 ");
        pia.perfoml();
        System.out.println("Performance 2nd method of the car 1 "+=>+ pia.perfol());
        System.out.println("Driving time of the car 1 ");
    }
}
```

```

Output - Car (run) x
run:
Enter the no of Rides for the car 1
5
Enter the no of Rides for the car 2
6
Enter the Fuel Capacity of the car 1
20
Enter the Fuel Capacity of the car 2
26
Enter the Fuel Consumption of the car 1
12
Enter the Fuel Consumption of the car 2
14

No of Rides in the car 1 =>5
Total Fuel Capacity of the car 1 =>20.0
Fuel Consumption of the car 1 =>12.0
Performance 1st method of the car 1
Performance 1.6666666666666667
Performance 2nd method of the car 1 =>1.66
Driving time of the car 1 =>42.0

No of Rides in the car 2 =>6
Total Fuel Capacity of the car 2 =>26.0
Fuel Consumption of the car 2 =>14.0
Performance 1st method of the car2
Performance 1.8571428571428572
Performance 2nd method of the car 2 =>1.66
Driving time of the car 2 =>42.0
BUILD SUCCESSFUL (total time: 16 seconds)

StartPage x Car.java x
Source History
pia.perform1();
System.out.println("Performance 2nd method of the car 1 "+=>"+ pia.perfol() );
System.out.println("Driving time of the car 1 "+=>"+ pia.ct1(3.5) );

System.out.println("");

System.out.println("No of Rides in the car 2 "+=>"+obj.ride2);
System.out.println("Total Fuel Capacity of the car 2 "+=>"+obj.fca2);
System.out.println("Fuel Consumption of the car 2 "+=>"+obj.fc2);
System.out.println("Performance 1st method of the car2 ");
obj.perform();
System.out.println("Performance 2nd method of the car 2 "+=>"+ pia.perfol() );
System.out.println("Driving time of the car 2 "+=>"+ pia.ct1(3.5) );

}

}

class Car1
{
    int ride1;
    double fca1;
    double fc1;

    void perform1()
    {
        double p;
        p= fca1/fc1;
        System.out.println("Performance "+p);
    }

    double perfol()
    {
        double p;
        p= fca1/fc1;
    }
}

```

```

Output - Car (run) x
run:
Enter the no of Rides for the car 1
5
Enter the no of Rides for the car 2
6
Enter the Fuel Capacity of the car 1
20
Enter the Fuel Capacity of the car 2
26
Enter the Fuel Consumption of the car 1
12
Enter the Fuel Consumption of the car 2
14

No of Rides in the car 1 =>5
Total Fuel Capacity of the car 1 =>20.0
Fuel Consumption of the car 1 =>12.0
Performance 1st method of the car 1
Performance 1.6666666666666667
Performance 2nd method of the car 1 =>1.66
Driving time of the car 1 =>42.0

No of Rides in the car 2 =>6
Total Fuel Capacity of the car 2 =>26.0
Fuel Consumption of the car 2 =>14.0
Performance 1st method of the car2
Performance 1.8571428571428572
Performance 2nd method of the car 2 =>1.66
Driving time of the car 2 =>42.0
BUILD SUCCESSFUL (total time: 16 seconds)

StartPage x Car.java x
Source History
double perfol()
{
    double p;
    p= fca1/fc1;
    return p;
}

double ct1(double t)
{
    return fca1*t ;
}

}

class Car2
{
    int ride2;
    double fca2;
    double fc2;

    void perform()
    {
        double p;
        p= fca2/fc2;
        System.out.println("Performance "+p);
    }

    double perfo()
    {
        double p;
        p= fca2/fc2;
        return p;
    }

    double ft(double t)

```

The screenshot shows an IDE window with two panes. The left pane, titled "Output - Car (run)", displays the execution output of a Java program. The right pane, titled "Source", shows the source code for "Car.java".

Output - Car (run):

```
run:
Enter the no of Rides for the car 1
5
Enter the no of Rides for the car 2
6
Enter the Fuel Capacity of the car 1
20
Enter the Fuel Capacity of the car 2
26
Enter the Fuel Consumption of the car 1
12
Enter the Fuel Consumption of the car 2
14

No of Rides in the car 1 =>5
Total Fuel Capacity of the car 1 =>20.0
Fuel Consumption of the car 1 =>12.0
Performance 1st method of the car 1
Performance 1.6666666666666667
Performance 2nd method of the car 1 =>1.66
Driving time of the car 1 =>42.0

No of Rides in the car 2 =>6
Total Fuel Capacity of the car 2 =>26.0
Fuel Consumption of the car 2 =>14.0
Performance 1st method of the car2
Performance 1.8571428571428572
Performance 2nd method of the car 2 =>1.66
Driving time of the car 2 =>42.0
BUILD SUCCESSFUL (total time: 16 seconds)
```

Source: Car.java

```
double fca2;
double fc2;

void perform()
{
    double p;
    p= fca2/fc2;
    System.out.println("Performance "+p);
}

double perfo()
{
    double p;
    p= fca2/fc2;
    return p;
}

double ft(double t)
{
    return fc2*t ;
}
```

At the bottom right of the IDE, there is a watermark: "Activate Windows Go to Settings to activate Windows."