PROJECT ON CAR RENT

WE USE TOPIC IN THIS PROJECT WHICH ARE FOLLOWING :

1). User input – output :-

- Syntax :- In Java, the primary way to obtain user input from the console is by utilizing the Scanner class, which is part of the java.util package.

import java.util.Scanner;

Create a Scanner object: Instantiate a Scanner object, typically linking it to System.in, which represents the standard input stream (console).

Scanner scanner = new Scanner(System.in);

```
import java.util.Scanner;


public class carbooking {
    public static void main(String[] args) {
        System.out.print("Do you want take a car on the rent : ");
        Scanner n = new Scanner(System.in);
            String str = n.next();
```

2). Print Statements :-

- Syntax :- This method prints the specified data to the console but does move the cursor to the next line

`System.out.println()` (Print Line)

This method prints the specified data to the console but does not move the cursor to the next line

`System.out.print)` (Print)

```
System.out.println("Welcome to my car rent showroom");
System.out.println("I have five car of the different modals for the rent with full details : \n");
System.out.println("(1). Car-id = 123\t\t\t\t\t\tModel = Maruti Dzire\t\t\t\t\tCar-name = 2020 Maruti Suzuki DZire VXi AMT\nEngine = 1197 cc, 4 Cylinders Inline, 4 Valve
System.out.println("(2). Car-id = 134\t\tModel = Honda CR-V\t\tCar-name = 2019 Honda CR-V 2.0\nEngine = 1597 cc - 2354 cc\tPower = 118.3 - 187.4 bhp\nTorque = 350 Nm - 3
System.out.println("(3). Car-id = 203\t\t\tModel = Toyota Camry\t\t\tCar-name = 2016 Toyota Camry Hybrid\nEngine = 2487 cc\t\t\tFuel Type = Hybrid (Electric + Petrol)\nT
System.out.println("(4). Car-id = 145\t\t\t\t\t\tModel = Hyundai Aura\t\t\tCar-name = 2022 Hyundai Aura S 1.2 Petrol\nEngine = 1197 cc, 4 Cylinders Inline, 4 Valves/Cyli
System.out.println("(5). Car-id = 590\tModel Toyota Corolla\tCar-name = 2013 Toyota Corolla Fz\nFuel Type = Petrol\tTransmission = Manual\n");
```

3). Apply while loop() Statement :-

- Syntax :- while (condition) {

  // Code to be executed repeatedly

  // Update statements for loop control variables

  }

Example :-

Java

```java
public class WhileLoopExample {
    public static void main(String[] args) {
        int count = 0; // Initialization of loop control variable

        while (count < 5) { // Condition: loop continues as long as count is les
            System.out.println("Count is: " + count); // Code to be executed
            count++; // Update statement: increments count in each iteration
        }
        System.out.println("Loop finished. Final count: " + count);
    }
}
```

```
while("Yes".equals(str) || "yes".equals(str) || "YES".equals(str))
{
    System.out.println("Welcome to my car rent showroom");
    System.out.println("I have five car of the different modals for the rent with full details : \n");
    System.out.println("(1). Car-id = 123\t\t\t\t\t\tModel = Maruti Dzire\t\t\t\t\tCar-name = 2020 Maruti Suzuki DZire VXi AMT\nEngine = 1197 cc, 4 Cylinders I
    System.out.println("(2). Car-id = 134\t\tModel = Honda CR-V\t\tCar-name = 2019 Honda CR-V 2.0\nEngine = 1597 cc - 2354 cc\tPower = 118.3 - 187.4 bhp\nTorqu
    System.out.println("(3). Car-id = 203\t\t\tModel = Toyota Camry\t\t\tCar-name = 2016 Toyota Camry Hybrid\nEngine = 2487 cc\t\t\tFuel Type = Hybrid (Electri
    System.out.println("(4). Car-id = 145\t\t\t\t\tModel = Hyundai Aura\t\t\tCar-name = 2022 Hyundai Aura S 1.2 Petrol\nEngine = 1197 cc, 4 Cylinders Inline,
    System.out.println("(5). Car-id = 590\tModel Toyota Corolla\tCar-name = 2013 Toyota Corolla Fz\nFuel Type = Petrol\tTransmission = Manual\n");
    System.out.print("Which car do you like for the rent : ");
    Scanner choices = new Scanner(System.in);
    String str1 = choices.nextLine();
    if("2020 Maruti Suzuki DZire VXi AMT".equals(str1) || "1".equals(str)){
        System.out.println("Per day price is = ₹2,000 - ₹3,000");
    }
    else if("2019 Honda CR-V 2.0".equals(str1) || "2".equals(str1)){
        System.out.println("Per day price is = ₹2,000 - ₹3,500");
    }
    else if("2016 Toyota Camry Hybrid".equals(str1) || "2013 Toyota Corolla Fz".equals(str1) || "3".equals(str1) || "4".equals(str1)){
```

```
while("NO".equals(str2) || "no".equals(str2)){
    System.out.println("Thanks for visiting my showroom");
    break;
```

4). Use if - else statements : -

- If statement Syntax :-

  If (condition)

  {

  //Statements to be executed

  }

- else –if statement Syntax :-

```
else if (condition2) {
        // Code block to be executed if condition1 is false AND
condition2 is true
}
```

- else statement Syntax :-

```
else {
        // Code block to be executed if the condition is false
}
```

Example :-

```java
public class NestedIfElseCondition {
    public static void main(String[] args) {
        double totalMarks = 382;
        double perc = (totalMarks / 500) * 100;
        String grade; // Declaring grade variable outside if-else blocks

        if (perc >= 80) {
            grade = "A+";
        } else if ((perc >= 70) && (perc < 80)) {
            grade = "A";
        } else if ((perc >= 60) && (perc < 70)) {
            grade = "B+";
        } else {
            grade = "B";
        }

        System.out.println("The percentage of the student is: " + perc);
        System.out.println("\nThe grade of the student is: " + grade);
    }
}
```

Run Code >>

```java
if("2020 Maruti Suzuki DZire VXi AMT".equals(str1) || "1".equals(str)){
    System.out.println("Per day price is = ₹2,000 - ₹3,000");
}
else if("2019 Honda CR-V 2.0".equals(str1) || "2".equals(str1)){
    System.out.println("Per day price is = ₹2,000 - ₹3,500");
}
else if("2016 Toyota Camry Hybrid".equals(str1) || "2013 Toyota Corolla Fz".equals(str1) || "3".equals(str1) || "4".equals(str1)){
    System.out.println("Per day price is = ₹3,500 - ₹4,000");
    }
else if("2022 Hyundai Aura S 1.2 Petrol".equals(str1) || "5".equals(str1)){
    System.out.println("Per day price is =  ₹3,000 - ₹5,250");
}
    System.out.println();
System.out.print("Do you want carry more information : ");
Scanner sc = new Scanner(System.in);
String str2 = sc.next();
if("YES".equals(str2) || "yes".equals(str2) || "Yes".equals(str2) || "ya".equals(str2) ){
    System.out.print("How much time you will take the car for rent : ");
    Scanner x = new Scanner(System.in);
    byte n1 = x .nextByte();
    if(n1 >= 5 && n1 < 10){
        System.out.println("5% discount at the total price");
    }
    else if(n1 >= 10 && n1 <15){
        System.out.println("12% discount at the total price");
    }
```

5). Use of break and continue statements :-

- Syntax of break() statement :-

  The **`break`** statement in Java is used to terminate the execution of a loop (e.g., **`for`**, **`while`**, **`do-while`**) .

  When a **`break`** statement is encountered, control immediately transfers to the statement following the terminated construct.

- Syntax of continue() statement :-

  The **`continue`** statement in Java is used within loops (for, while, or do-while) to skip the current iteration and proceed to the next iteration of the loop.

Example :-

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    break;
  }
  System.out.println(i);
}
```

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    continue;
  }
  System.out.println(i);
}
```

```
    else if("No".equals(str3) || "no".equals(str3) || "NO".equals(str3))
    {
        System.out.println("Thanks for visiting my showroom");
        break;
    }
}
while("NO".equals(str2) || "no".equals(str2)){
    System.out.println("Thanks for visiting my showroom");
    break;
}
break;
```

6). Use of File Handling :-

- Syntax of File – Handling :-

  File Handling is an integral part of any programming language as file handling enables us to store the output of any particular program in a file and allows us to perform certain operations on it.

  In simple words, file handling means reading and writing data to a file.

- CREATE A FILE()
- WRITE ON A FILE()
- READ FROM A FILE()
- DELETE A FILE()

Creating an InputStream:

```
// Creating an InputStream
InputStream obj = new FileInputStream();
```

Creating an OutputStream:

```
// Creating an OutputStream
OutputStream obj = new FileOutputStream();
```