Experiment No. 3
Implement a program that demonstrates the concepts of class and objects
Date of Performance:
Date of Submission:

Aim: Implement a program that demonstrates the concepts of class and objects

Objective: To develop the ability of converting real time entity into objects and create their classes.

Theory:

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties i.e., members and 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.
2. class keyword: class keyword is used to create a class.
3. Class name: The name should begin with a initial letter (capitalized by convention).
4. 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.
5. 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.
6. Body: The class body surrounded by braces, {}.

An OBJECT is a basic unit of Object-Oriented Programming and represents the real-life entities. A typical Java program creates many objects, which 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.

Code:

```java
import java.io.*;
class Car {
    String brand;
    String color;
    void displayInfo() {
        System.out.println("Brand: " + brand);
        System.out.println("Color: " + color);
    }
}

public class Mclaren{
    public static void main(String args[]) {
        Car car1 = new Car();
        Car car2 = new Car();
        Car car3 = new Car();

        car1.brand = "Scorpio";
        car1.color = "Black";

        car2.brand = "BMW";
        car2.color = "Black";

        car3.brand = "Ferrari";
        car3.color = "Red";
        System.out.println("Car 1 Information:");
        car1.displayInfo();

        System.out.println("\nCar 2 Information:");
        car2.displayInfo();

        System.out.println("\nCar 3 Information:");
        car3.displayInfo();

    }
}
```

OUTPUT

Output

```
java -cp /tmp/CaVycfE37E/Mclaren
Car 1 Information:
Brand: Scorpio
Color: Black

Car 2 Information:
Brand: BMW
Color: Black

Car 3 Information:
Brand: Ferrari
Color: Red

=== Code Execution Successful ===
```

Conclusion:

- **Class Template**: Defines the structure and behavior (fields and methods) of objects.

- **Objects**: Instances of the class, created using the new keyword and initialized with specific values.

This approach encapsulates data and behavior, promoting code reuse and modularity.