

Practical No.1

Aim: Write a Java program to implement class and object

Resources Required:

- Java Development Kit (JDK)
- Text Editor (e.g., Notepad++) or IDE (e.g., Eclipse, IntelliJ IDEA, NetBeans)
- Command-line terminal or Java compiler

Theory:

In Java, a **class** is a blueprint for creating objects. A **class** contains fields (variables) and methods (functions) that define the behavior of an object. An **object** is an instance of a class.

Key Concepts:

- A class defines structure and behavior.
- An object is created using the `new` keyword.
- Objects can access fields and methods of the class.

Example:

If you define a class called `Car`, you can create multiple `Car` objects with different properties.

Class Syntax

```
class ClassName {  
    // Fields (variables)  
    dataType variableName;  
  
    // Methods  
    returnType methodName() {  
        // method body  
    }  
}
```

Object Syntax:

```
ClassName objectName = new ClassName();
```

Source Code:

// Class definition

```
class Student {  
    // Data members (fields)  
    String name;  
    int age;
```

```
// Method to display student details
void displayInfo() {
    System.out.println("Name: " + name);
    System.out.println("Age: " + age);
}
}
```

```
// Main class
public class Main {
    public static void main(String[] args) {
        // Creating an object of Student
        Student student1 = new Student();

        // Assigning values to object
        student1.name = "Alice";
        student1.age = 20;

        // Calling method using object
        student1.displayInfo();
    }
}
```

Output:

Name: Alice
Age: 20

Conclusion:

The Java program successfully demonstrates the implementation of class and object. We defined a class Student with fields and methods, then created an object student1 to access those members. This illustrates the object-oriented concept of encapsulation in Java.