



Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

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



Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

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:

```
class Person {

    String name;
    int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

public class ClassAndObjectExample {
    public static void main(String[] args) {

        Person person1 = new Person("Mahek Shah", 19);

        Person person2 = new Person("Bob", 30);

        System.out.println("Person 1:");
        person1.displayInfo();

        System.out.println("\nPerson 2:");
        person2.displayInfo();
    }
}
```



```
C:\Mahek Shah CSEDS>javac ClassAndObjectExample.java
```

```
C:\Mahek Shah CSEDS>java ClassAndObjectExample
```

```
Person 1:
```

```
Name: Mahek Shah
```

```
Age: 19
```

```
Person 2:
```

```
Name: Bob
```

```
Age: 30
```

Conclusion:

In Java programming, classes and objects are foundational concepts that are integral to the object-oriented programming paradigm. Java is often regarded as a pure object-oriented language because nearly everything in Java is represented as an object. These concepts play a pivotal role in the Java programming landscape, enabling effective code organization, reusability, and the structured representation of real-world entities. The use of classes and objects in Java is not only a core architectural principle but also a practical approach for creating, modeling, and managing software components and data structures in a systematic and efficient manner.