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:

```
class Student
 {
int id;
 String name;    public static void
main(String args[])
 {
 Student s1 = new Student();
 System.out.println(s1.id);
 System.out.println(s1.name);
 }
 }
```

OUTConclusion:

creating a class template involves defining the structure and behavior of a class. This includes declaring variables, defining methods, and specifying any necessary constructors. Once the class template is defined, we can create objects of that class by using the "new" keyword followed by the class name and any required constructor arguments. These objects can then be used to access the variables and methods defined within the class template, allowing us to work with and manipulate data in a more organized and modular way. By creating class templates and their objects, we can effectively model real-world entities and implement reusable code in our Java programs.